



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Tecnologia

**Bruna do Nascimento Benevides e Thiago Feitosa**

**Protótipo de um sistema hidráulico para detecção de  
possíveis vazamentos em tubulações com válvula  
automatizada**

Limeira  
2023

**Bruna do Nascimento Benevides e Thiago Feitosa**

**Protótipo de um sistema hidráulico para detecção de possíveis vazamentos em tubulações com válvula automatizada**

Dissertação apresentada à Faculdade de Tecnologia da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Telecomunicações.

**Orientadora: Profa. Dra. Talía de Simões dos Santos Ximenes**  
**Coorientadora: Prof. Dra. Laura Maria Canno Ferreira Fais**

Este trabalho corresponde à versão final da Dissertação defendida por Bruna do Nascimento Benevides e Thiago Feitosa e orientada pela Profa. Dra. Talía de Simões dos Santos Ximenes.

Limeira  
2023

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Faculdade de Tecnologia  
Mariana Xavier - CRB 8/9615

B435p Benevides, Bruna do Nascimento, 1998-  
Protótipo de um sistema hidráulico para detecção de possíveis vazamentos em tubulações com válvula automatizada / Bruna do Nascimento Benevides, Thiago Feitosa. – Limeira, SP : [s.n.], 2023.

Orientador: Talía Simões dos Santos Ximenes.  
Coorientador: Laura Maria Canno Ferreira Fais.  
Trabalho de Conclusão de Curso (graduação) – Universidade Estadual de Campinas, Faculdade de Tecnologia.

1. Estruturas hidráulicas. 2. Detectores de vazamento. I. Feitosa, Thiago, 1998-. II. Ximenes, Talía Simões dos Santos, 1980-. III. Fais, Laura Maria Canno Ferreira, 1977-. IV. Universidade Estadual de Campinas. Faculdade de Tecnologia. V. Título.

Informações adicionais, complementares

**Título em outro idioma:** Prototype of a hydraulic system for detecting possible leaks in pipeline witherspoon automated valve

**Palavras-chave em inglês:**

Hydraulic structures

Leak detectors

**Titulação:** Engenheira de Telecomunicações

**Banca examinadora:**

Talía Simões dos Santos Ximenes [Orientador]

Luis Fernando de Avila

Marcela Cravo Ferreira

**Data de entrega do trabalho definitivo:** 12-12-2023

## FOLHA DE APROVAÇÃO

Abaixo se apresentam os membros da comissão julgadora da sessão pública de defesa de dissertação para o Título de Bacharel em Engenharia de Telecomunicações na área de concentração , a que se submeteu o aluno Bruna do Nascimento Benevides e Thiago Feitosa, em 12 de dezembro de 2023 na Faculdade de Tecnologia – FT/UNICAMP, em Limeira/SP.

**Profa. Dra. Talia de Simões dos Santos Ximenes**

Presidente da Comissão Julgadora

**Prof. Dra. Laura Maria Canno Ferreira Fais**

FT/UNICAMP

**Prof. Dr. Luis Fernando De Ávila**

FT/UNICAMP

**Profa. Dra. Marcela Cravo Ferreira**

FT/UNICAMP

Ata da defesa, assinada pelos membros da Comissão Examinadora, encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Graduação da Faculdade de Tecnologia.

# Agradecimentos

Agradecemos aos nossos familiares, que são a primeira base e estrutura de nossas formações como cidadãos, nos ensinando valores éticos e morais, valores estes que levamos sempre conosco, são nossos exemplos, nos fazem enxergar o potencial e possibilidade em situações distintas, expondo sempre o nosso melhor para com os diferentes desafios da vida. Não seríamos nada sem o apoio, exemplo, dedicação e ensinamentos passados ao longo do tempo.

Este trabalho significa e reúne um conjunto de esforços mútuos entre nós (Bruna do Nascimento Benevides e Thiago Feitosa). Onde enxergamos que a engenharia de telecomunicações é um curso extremamente multidisciplinar, nos fazendo traçar um paralelo entre as nossas vontades por meio deste trabalho e as aplicações distintas dentro da sociedade. Uma vez que, enxergamos o potencial intrusivo das tecnologias mediante ao grande cenário atual, que solicita de aplicações dos meios tecnológicos para angariar e assegurar um mundo melhor para todos.

Agradecemos aos nossos amigos que fizeram dos nossos dias na faculdade um lugar melhor, conseguimos aprender e conviver em paz dentre a “correria” do ambiente das matérias, dificuldades e desafios, sempre tínhamos uns aos outros. É reconfortante saber que o esforço mútuo entre um grupo transforma um ambiente, e este se modifica em um lugar melhor. Entendendo que todos tem potencial e existe espaço para todos crescerem e desenvolverem suas habilidades e competências. Amigo é uma palavra extremamente complexa, levaremos com carinho cada um que esteve conosco, direta e indiretamente. um sincero agradecimento a cada um destes seres que tiveram empatia para com o aprendizado alheio, deixando de olhar para si e querendo o melhor para quem esteve junto, por algum período. Aos técnicos do laboratório Rodrigo Luiz Ximenes e João Francisco Viana.

A nossa orientadora, que enxergou e confiou potencial em nós, nos deixou livres para sermos criativos e desenvolvermos um trabalho multidisciplinar. Nos proporcionando orientação, suporte e conhecimento. Compartilhando ao longo de todo o processo de desenvolvimento suas experiências e feedbacks para este trabalho, sendo fundamental. Queremos deixar claro nossa imensa gratidão por toda a dedicação, paixão e comprometimento que você demonstrou ao longo de todo nosso tempo juntos. Você é uma professora verdadeiramente incrível, e seu impacto em nossas vidas e aprendizados é inestimável. Além disso, sua capacidade de tornar as aulas interessantes e desafiadoras ao mesmo tempo é notável. Você nos incentivou a pensar criticamente, a questionar e a explorar o mundo ao nosso redor. Essas lições não se limitam apenas à sala de aula, elas moldam nossa visão de mundo e nossa sede de conhecimento.

Eu Thiago Feitosa quero agradecer em especial a senhora Elizabete Cristina Ciriaco Feitosa. Minha melhor amiga. Uma mulher guerreira, batalhadora que sempre correu atrás dos seus objetivos para possibilitar o melhor para sua família. Muita das vezes deixando de olhar pra si mesma para melhor cuidar de quem está em volta. Agradeço a senhora mãe.

Agradecemos a todos os envolvidos! Obrigado.

# Resumo

Este trabalho tem como objetivo a proposta de um sistema de baixo custo para monitoramento de uma rede de água, a fim de detectar vazamentos e acionar uma válvula que fecha o sistema. Os testes foram realizados no Laboratório de Hidráulica e Mecânica dos Fluidos da Faculdade de Engenharia Civil e Urbanismo da Universidade Estadual de Campinas (FECFAU/UNICAMP), em uma rede simulada. Para os testes, foram utilizados sensores para medição de vazão, nível d'água, ruído, turbidez, temperatura e um extensômetro, para leitura em tempo real. As leituras foram recebidas e processadas por uma placa microcontroladora (Arduino UNO) e por meio de um módulo Wi-Fi (ESP8266), os dados enviados para uma página da Web (linguagem PHP) para visualização dos resultados dos sensores. Em testes de laboratório, os sensores se mostraram adequados para detectar vazamentos na rede e disparar alarmes sonoros e visuais em caso de vazamentos e fechar a válvula de alimentação do sistema. O sistema pode detectar um vazamento e a automatização da válvula, visando o sensoriamento e a detecção de vazamentos, bem como a prevenção de rupturas em tubulações de água. A válvula automatizada é controlada por meio de uma página web, oferecendo facilidade e flexibilidade ao usuário, sendo uma solução promissora para a gestão eficiente da água.

# Abstract

This work aims to propose a low-cost system for monitoring a water network to detect leaks and activate a valve that shuts off the system. Tests were conducted in the Laboratory of Hydraulics and Fluid Mechanics at the School of Civil and Urban Engineering of the State University of Campinas (FECFAU/UNICAMP) in a simulated network. For the tests, sensors were used to measure flow, water level, noise, turbidity, temperature, and a strain gauge for real-time reading. Readings were received and processed by a microcontroller board (Arduino UNO) and via a Wi-Fi module (ESP8266), with data sent to a web page (PHP language) for sensor results visualization. In laboratory tests, the sensors proved suitable for detecting leaks in the network, triggering audible and visual alarms in case of leaks, and closing the system's supply valve. The system can detect a leak, and valve automation, aiming for leak sensing and detection, as well as the prevention of water pipe ruptures. The automated valve is controlled through web page, offering ease and flexibility to the user, making it a promising solution for efficient water management.

# Lista de Figuras

3.1	Placa Arduíno UNO . . . . .	19
3.2	Placa Esp8266 . . . . .	21
3.3	Placa ESP8266 Pinagem . . . . .	22
3.4	Motor de Passo . . . . .	23
3.5	Diagrama Elétrico do Motor de Passo . . . . .	24
3.6	Steps Motor de Passo . . . . .	24
3.7	Extensômetro . . . . .	25
3.8	Buzzer . . . . .	26
3.9	Display LCD . . . . .	27
3.10	Relé . . . . .	28
3.11	Transistor . . . . .	29
3.12	Circuito microfone de eletreto . . . . .	30
3.13	Figura Ilustrativa Registro de Pressão . . . . .	31
3.14	Desenho Técnico Registro de Pressão . . . . .	31
3.15	Layout do circuito do Drive. . . . .	32
3.16	Sensor de temperatura . . . . .	34
3.17	Sensor de Fluxo de Água 1/2 YF-S201B . . . . .	35
3.18	Sensor de nível . . . . .	36
3.19	Sensor de nível . . . . .	37
3.20	Sensor de nível . . . . .	37
3.21	Sensor de turbidez . . . . .	38
3.22	Protótipo do sistema hidráulico . . . . .	41
3.23	Protótipo do sistema hidráulico com destaque aos sensores utilizados . . . . .	42
3.24	Página web . . . . .	43
3.25	pagina web . . . . .	43
3.26	Acoplamento da bucha com o registro e o motor . . . . .	44
3.27	Sistema Inicial montado na protoboard e Arduíno . . . . .	45
3.28	Layout Circuito no Eagle; Prensa Térmica ; Placa PCV com circuito . . . . .	48
3.29	Perfuração; ; Placa PCV com circuito . . . . .	50
4.1	Demonstração na página web para sensor de vazão. . . . .	51
4.2	Demonstração na página web para sensor de temperatura. . . . .	52
4.3	Composição do sensor extensometro. . . . .	52
4.4	Demonstração na página web para sensor de som. . . . .	53
4.5	Demonstração na página web para sensor de turbidez. . . . .	53
4.6	Demonstração na página web para sensor de nível. . . . .	53
4.7	página web . . . . .	54
4.8	Monitor serial . . . . .	55
4.9	Alerta de fechamento e abertura da válvula na página web . . . . .	56

# Lista de Tabelas

1.1	Estimativa da distribuição global de água no mundo . . . . .	13
3.1	Característica da placa microcontroladora Arduino UNO . . . . .	20
3.2	Especificações da Placa ESP8266 NodeMCU . . . . .	21
3.3	Especificações Motor de Passo utilizado . . . . .	23
3.4	Ficha Técnica Extensômetro . . . . .	25
3.5	Ficha Técnica Buzzer . . . . .	26
3.6	Ficha Técnica Relé . . . . .	28
3.7	Ficha Técnica Microfone de Eletreto . . . . .	29
3.8	Ficha Técnica Registro de Pressão DECA . . . . .	32
3.9	Ficha Técnica Sensor de Temperatura . . . . .	34
3.10	Especificação Sensor de Fluxo de Água 1/2 YF-S201B . . . . .	36
3.11	Especificações sensor de turbidez . . . . .	38

# Lista de Abreviaturas e Siglas

IoT	Internet das Coisas
MCA	Metros de Coluna D'Água
OMS	Organização Mundial de Saúde
PCB	Printed Circuit Board
VCC	Tensão de Alimentação
VRM	Tensão Reversa de Pico não repetitiva
VRRM	Tensão Reversa Média

# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Distribuição da Água . . . . .	12
1.2	Automatização de Dispositivo Hidráulico . . . . .	14
1.3	Possíveis utilizações de Registros Automatizados . . . . .	15
<b>2</b>	<b>Objetivo</b>	<b>17</b>
2.1	Objetivos Específicos . . . . .	17
<b>3</b>	<b>Desenvolvimento</b>	<b>18</b>
3.1	Materiais . . . . .	18
3.1.1	Arduíno . . . . .	18
3.1.2	ESP8266 . . . . .	20
3.1.3	Motor de Passo . . . . .	22
3.1.4	Extensômetro . . . . .	24
3.1.5	Buzzer . . . . .	25
3.1.6	Display LCD . . . . .	27
3.1.7	Relé . . . . .	28
3.1.8	Transistor . . . . .	28
3.1.9	Sensor de Som / Detecção de Ruídos . . . . .	29
3.1.10	Registro de Pressão . . . . .	30
3.1.11	Drive . . . . .	30
3.1.12	Sensor de Temperatura . . . . .	33
3.1.13	Sensor de Fluxo / Sensor de Vazão . . . . .	35
3.1.14	Sensor de nível de água . . . . .	35
3.1.15	Sensor de turbidez . . . . .	37
3.2	Métodos . . . . .	38
3.2.1	Protótipo do Sistema Hidráulico . . . . .	38
3.2.2	Página web . . . . .	42
3.2.3	Conexão Válvula e Motor de Passo . . . . .	44
3.2.4	Montagem e código dos Circuitos do Sistema . . . . .	44
3.2.5	Desenvolvimento Placa PCB . . . . .	48
3.2.6	Testes Realizados em Sistemas Hidráulicos . . . . .	49
<b>4</b>	<b>Resultados e Discussões</b>	<b>51</b>
4.1	Conectividade e Transmissão de Dados . . . . .	53
<b>5</b>	<b>Conclusões</b>	<b>57</b>
	<b>Referências Bibliográficas</b>	<b>59</b>



# Capítulo 1

## Introdução

### 1.1 Distribuição da Água

É de compreensão universal que a subsistência humana assim como o progresso da população depende essencialmente da água. Segundo a Organização Mundial de Saúde (OMS), o abastecimento de água insere-se no conceito mais amplo de saneamento básico, visando garantir condições de salubridade que protejam a saúde. Com o crescimento da população, e a demanda requerida pela água, sabe-se que sua disponibilidade vem sendo insuficiente em algumas regiões do planeta e esse fato tem se agravado com o passar dos anos.

Neste cenário, é essencial que o fornecimento de água seja realizado por instalações que sejam aptas a proporcionar esse recurso com qualidade, constância e de forma acessível, analisando e considerando as futuras gerações, respeitando os interesses da população que também utilizam desses mananciais. Desta forma, as pessoas responsáveis por todo o planejamento, implementação e manutenção das instalações de abastecimento de água, devem ter consciência desses fatos e também considerá-los (HELLER; PÁDUA, 2010).

Estima-se que 97,5% da água do planeta Terra é água salgada, e apenas 2,6% é de água doce. Da quantidade de água doce disponível, aproximadamente 68,7% está em calotas polares, geleiras e neve permanente, logo não estão disponíveis para consumo (SCHOOL, 2019).

A Tabela 1.1 mostra uma estimativa da distribuição global de água no mundo. Pode-se observar que do total de água doce, os outros 30% aproximadamente da água doce são águas subterrâneas. Os rios são a fonte mais utilizada da água doce de superfície, porém constitui apenas 0.007% da água total.

Tabela 1.1: Estimativa da distribuição global de água no mundo (GLEICK, 1993).

	Área de distribuição (103km)	Volume (103km)	Camada (m)	Porcentagem das reservas globais de água total	Porcentagem das reservas globais de água doce
<b>Oceano no mundo</b>	361.300	1.338.00	3.700	96.5	-
<b>Água subterrânea</b>	134.800	23.400	174	1.7	-
Água doce		10.530	78	0.76	30.1
Umidade do solo		16.5	0.2	0.001	0.05
<b>Geleiras e cobertura de neve permanente</b>	16.227	24.064	1.463	1.74	68.7
Antártico	13.980	21.600	1.546	1.56	61.7
Groenlândia	1.802	2,340	1.298	0.17	6.68
Ilhas do Ártico	226	83.5	369	0.006	0.24
Regiões montanhosas	224	40.6	181	0.003	0.12
<b>Gelo Terrestre / Permafrost</b>	21.000	300	14	0.022	0.86
<b>Reservas de água nos lagos</b>	2.058.7	176.4	85.7	0.0013	-
Água doce	1.236.4	91	73.6	0.007	0.26
Água salgada	822.3	85.4	103.8	0.006	-
<b>Água do pântano</b>	2.682.6	11.47	4.28	0.0008	0.03
<b>Fluxo do Rio</b>	148.800	2.12	0.014	0.0002	0,006
<b>Água biológica</b>	510.000	1.12	0.002	0.0001	0.003
<b>Água atmosférica</b>	510.000	12.9	0.0.25	0.001	0.04
<b>Reservas totais de água</b>	510.000	1.385.984	2.718	100	-
<b>Reservas totais de água doce</b>	148.800	35.029	235	2.53	100

Segundo dados do Ministério do Desenvolvimento Regional Secretaria Nacional de Saneamento (2021), indicam que o índice de perdas na distribuição de água representa 40,1%, isso significa que 59,9 litros são utilizados a cada 100 litros disponibilizados pelas prestadoras. No período de 2012 a 2015, esse indicativo chegou a ficar abaixo de 37%, ou seja, houve um crescimento de desperdício de água entre 2015 a 2020.

Essas perdas estão relacionadas a dois fatores, no caso de quando a água consumida não está sendo contabilizada (ligações clandestinas, gatos, falta de calibragem nos hidrômetros) chamada de perda aparente, e quando há vazamentos nas estruturas de distribuição, que é a perda real (MINISTÉRIO DO DESENVOLVIMENTO REGIONAL SECRETARIA NACIONAL DE SANEAMENTO, 2021).

## **1.2 Automatização de Dispositivo Hidráulico**

A crescente evolução da internet e o envolvimento com as redes sociais, tem transformado a forma que a humanidade se comunicam, interage e compartilham informações. A Internet das Coisas (IoT) é um conceito que se refere à interconexão de dispositivos físicos à internet, criando um ambiente inteligente onde os dispositivos podem coletar dados, trocar informações, e executar comandos. IoT permite a integração de sensores, possibilitando o sensoriamento das tubulações de água. Isso proporciona informações em tempo real sobre o consumo, distribuição e uso da água, facilitando a gestão hídrica e reduzindo o desperdício de água.

A crescente evolução da internet e avanço das redes sociais têm transformado a forma como a humanidade se comunica, interage e compartilha informações. Além disso, a Internet das Coisas (IoT) possibilita a interconexão de dispositivos físicos à internet, criando um ambiente inteligente onde os dispositivos podem coletar dados, trocar informações e acionar dispositivos. Isso é especialmente relevante no contexto da gestão hídrica, onde o IoT permite o sensoriamento das tubulações de água, fornecendo informações em tempo real sobre o consumo, distribuição e uso da água, resultando em uma melhor gestão e redução do desperdício. No âmbito industrial, a medição e controle de parâmetros como pressão, temperatura, vazão e nível desempenham um papel crucial.

A medição da vazão é especialmente importante, já que pequenas imprecisões podem ter um impacto significativo em termos de ganhos e perdas. Portanto, é essencial monitorar as variações nos fatores das tubulações, especialmente devido às perdas de vazão mencionadas no relatório SNIS-AE. Diante dessa necessidade, torna-se fundamental abordar a problemática da viabilidade de desenvolver um sistema eficiente e acessível. Esse sistema deve permitir o monitoramento contínuo e preciso dos parâmetros das tubulações, utilizando esse tipo de tecnologia para coletar e analisar os dados em tempo real, auxiliando na tomada de decisões e na implementação de medidas para redução de perdas e otimização da gestão hídrica.

### **1.3 Possíveis utilizações de Registros Automatizados**

A busca por maior eficiência, controle preciso e segurança tem impulsionado diversos setores, como agricultura, indústria e energia, a adotarem soluções de automatização em seus processos.

Na área da hidráulica, as automações nesse campo se destacam devido à importância da água em todos os processos. A irrigação na agricultura, por exemplo, permite que o Brasil seja um dos poucos países com a capacidade de triplicar sustentavelmente sua área irrigada (EMBRAPA, 2022). Dessa forma, a automatização de válvulas hidráulicas possibilita um manejo mais preciso da água, otimizando seu uso e aumentando a produtividade.

Nas indústrias, é importante destacar que vazamentos de água podem expor os funcionários a riscos significativos, principalmente quando ocorrem com pressões elevadas ou em altas temperaturas, podendo causar acidentes graves e comprometer a segurança dos trabalhadores. Além dos riscos humanos, os vazamentos também acarretam custos financeiros e perdas para as indústrias, seja pelo desperdício de recursos naturais, pela necessidade de reparos em equipamentos danificados ou pela interrupção da produção.

Além disso, a implementação de registros hidráulicos automatizados, que promovem a possibilidade de abri-los e fechá-los à longa distância em lugares de alta periculosidade, pode trazer benefícios significativos em termos de segurança e eficiência operacional. Em situações em que a intervenção humana direta pode ser arriscada ou inviável devido a riscos ambientais, periculosidades físicas ou condições adversas, essa opção se torna relevante.

Em resumo, a automação de válvulas hidráulicas oferece uma série de vantagens, pois a possibilidade de controlar os registros à distância permite uma resposta rápida e eficiente,

reduzindo o tempo de reação e minimizando os danos potenciais resultantes dessas situações críticas.

# Capítulo 2

## Objetivo

Este trabalho tem como objetivo desenvolver um protótipo de um sistema hidráulico em conjunto com a automatização de válvula integrado a uma página web, visando o sensoriamento e a detecção de vazamentos, bem como a prevenção de rupturas em tubulações de água.

### 2.1 Objetivos Específicos

1. Propor e desenvolver um protótipo de sistema hidráulico capaz de detectar através de sensores, de temperatura, extensômetro, som, fluxo, nível e turbidez a ocorrência de vazamentos no sistema.
2. Propor e desenvolver uma válvula automatizada com recursos de conectividade que permita a comunicação com página web com interface amigável, permitindo o acesso e o controle da válvula de forma remota e intuitiva.
3. Realizar testes e validações do sistema, avaliando seu desempenho, funcionalidades e usabilidade.
4. Realizar testes e análises do sistema desenvolvido, verificando sua eficácia.

# Capítulo 3

## Desenvolvimento

Aqui serão abordados todos os aspectos metodológicos da pesquisa realizada, descrevendo-se os procedimentos necessários e úteis para propor e desenvolver um sistema de automatização de válvula integrado a um aplicativo de rede social.

### 3.1 Materiais

#### 3.1.1 Arduíno

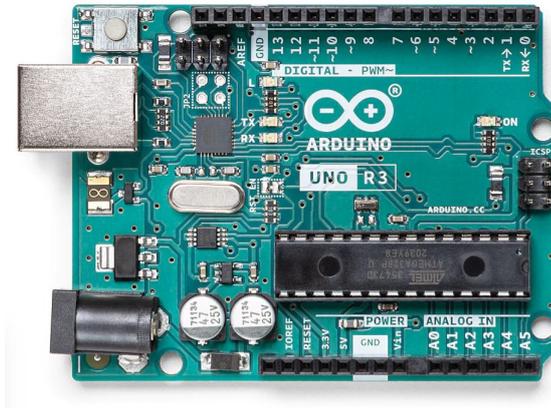
O Arduino é uma plataforma de computação física de código aberto que utiliza uma placa de entrada/saída simples e um ambiente de desenvolvimento baseado na linguagem Processing (BANZI, 2009).

A computação física envolve a criação de sistemas interativos físicos por meio da integração de software e hardware, permitindo que esses sistemas percebam e respondam ao mundo analógico (MULTILÓGICA-SHOP. (EQUIPE), 2023).

Isso envolve a concepção de objetos interativos capazes de interagir com seres humanos por meio de sensores e atuadores controlados por programas de software executados em microcontroladores, que são pequenos computadores integrados em um único chip. Desta forma, a Computação Física permite desenvolver projetos que unem eletrônica e programação para criar objetos que podem detectar e reagir ao ambiente ao redor, oferecendo uma experiência interativa para os usuários (BANZI, 2009).

O Arduino é composto pela a placa Arduino observada na Figura 3.1, que é o componente de hardware no qual é trabalhado para construir os objetos, e o ambiente de desenvolvimento, o Arduino IDE, que é o software que é realizado a execução em um computador. Estas são as

Figura 3.1: Placa Arduino UNO.



Fonte: (SILVEIRA, 2023)

principais partes do Arduino. A IDE é utilizada para criar um programa de computador, que é chamado de sketches (softwares escritos usando Arduino) que é carregado na placa Arduino, e esse programa diz à placa o deverpa ser feito (BANZI, 2009).

A placa Arduino é um dispositivo de microcontrolador compacto que incorpora um computador completo em um único chip. Existem diversas versões da Placa Arduino, mas a mais popular e também utilizada neste projeto é a Arduino UNO. A Tabela 3.1 mostra as especificações da placa microcontroladora Arduino Uno.

O Arduino Uno é uma placa de microcontrolador que utiliza o ATmega328P como base. Possui 14 pinos digitais que podem ser configurados como entradas ou saídas (sendo que 6 deles suportam saída PWM), 6 entradas analógicas, um ressonador cerâmico de 16 MHz, conexão USB, conector de energia, cabeçalho ICSP e um botão de reset. A placa contém todos os componentes necessários para o funcionamento do microcontrolador. Basta conectá-la a um computador através de um cabo USB ou alimentá-la com um adaptador AC-DC ou bateria para começar a utilizá-la.

A IDE (Ambiente Integrado de Desenvolvimento) do Arduino é um software executado no computador no qual permite escrever sketches para a placa Arduino. Essa IDE utiliza uma linguagem simples baseada na linguagem Processing.

Tabela 3.1: Característica da placa microcontroladora Arduino UNO

Microcontrolador	ATmega328P
Tensão operacional	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limite)	6-20V
Pinos de I/O digitais	14 (dos quais 6 fornecem saída PWM)
Pinos de E/S digitais PWM	6
Pinos de entrada analógica	6
Corrente CD por pino de I/O	20 mA
Corrente CD para pino de 3,3 V	50 mA
Memória flash	32 KB (ATmega328P), dos quais 0,5 KB são usados pelo bootloader.
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidade do relógio	16 MHz
LED_BUILTIN	13
Comprimento	68,6 mm
Largura	53,4 mm
Peso	25g

Fonte: (SILVEIRA, 2023)

### 3.1.2 ESP8266

O módulo Wi-Fi ESP8266 NodeMCU é uma placa de desenvolvimento que combina o chip ESP8266, uma interface usb-serial e um regulador de tensão 3,3V. A programação pode ser feita usando LUA ou a IDE do Arduino, utilizando a comunicação via cabo micro-usb.

A Figura 3.2 mostra a placa ESP8266 e a Figura 3.3 mostra a pinagem da placa.

A placa apresenta as seguintes especificações que podem ser observadas na tabela 3.2, conforme informações encontradas no site Eletrônicos (2016-2022).

Figura 3.2: Placa Esp8266



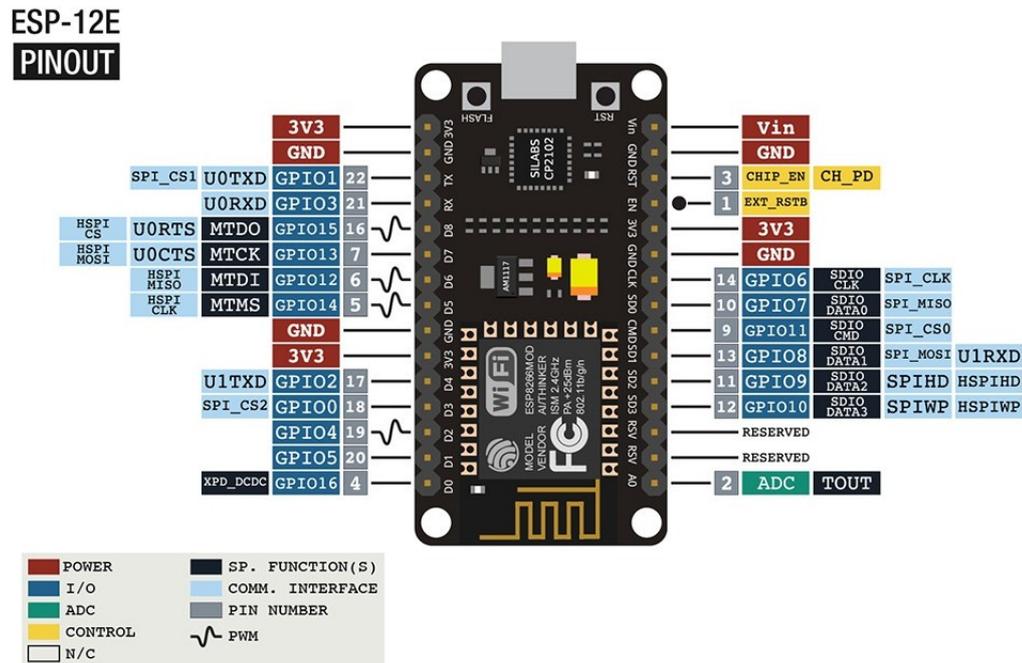
Fonte: (MULTIPEÇAS, 2023)

Tabela 3.2: Especificações da Placa ESP8266 NodeMCU

Módulo	ESP8266 ESP-12E
Conector micro-usb	Sim
Antena embutida	Sim
Tensão de Operação	4,5 à 9,0 VDC (Pino Vin)
Modos de operação	STA/AP/STA+AP
Suporta 5 conexões TCP/IP	Sim
Portas GPIO	11
Funções GPIO	PWM, I2C, SPI, etc;
WiFi 802.11 b/g/n	2.4 à 2.5 GHz
Taxa de transferência	110-460800 bps
Compatível com a IDE do Arduino	Sim
Suporta Upgrade remoto de firmware	Sim
Conversor analógico digital (ADC)	Sim
Dimensões	49 x 25,5 x 7 mm

Fonte : (MULTIPEÇAS, 2023)

Figura 3.3: Placa ESP8266 Pinagem



Fonte: (MULTIPEÇAS, 2023)

### 3.1.3 Motor de Passo

Motores de passo são dispositivos eletro-mecânicos que convertem pulsos elétricos em movimentos mecânicos precisos e controlados. Eles têm a capacidade de gerar variações angulares discretas, movendo o rotor ou eixo em incrementos conhecidos como "passos". O funcionamento desses motores depende da relação entre os impulsos elétricos recebidos e da sequência dos pulsos elétricos aplicados, refletindo diretamente a direção de rotação do motor (BRITES; SANTOS, 2008). A Figura 3.4 mostra uma ilustração do modelo do motor de passo utilizado para o projeto.

O controle de um motor de passo é realizado por meio da aplicação de pulsos elétricos nas bobinas do motor. Esses pulsos elétricos fazem com que o rotor do motor gire em uma posição específica, determinada pelo número de passos do motor. No caso deste projeto, conforme apresentado na Tabela 3.3, cada passo corresponde a um ângulo de 1,8 graus.

Figura 3.4: Motor de Passo



Fonte: (THOMSON, 2023)

Tabela 3.3: Especificações Motor de Passo utilizado

Modelo	Airpax 4SH-12A56S
Volts	12
Ohms / Bobina	21
Graus / Passo	1,8

Fonte: (THOMSON, 2023)

Cada bobina do motor de passo é composta por um certo número de enrolamentos de fio condutor. Conforme apresentado no Diagrama Elétrico do motor utilizado no projeto na Figura 3.5, o motor possui 4 bobinas.

O funcionamento dos passos do motor ocorre através da sequência de energização das bobinas, como mostrado na Figura 3.6. A energização específica de uma bobina resulta em um deslocamento angular discreto do rotor. Isso ocorre porque o rotor é magneticamente energizado. Ao polarizar sequencialmente as bobinas, há uma interação entre os campos magnéticos das bobinas e o rotor do motor, resultando no movimento do rotor em passos discretos (BRITES; SANTOS, 2008).

O diagrama do circuito na Figura 3.5 é construído com quatro chaves (Q1, Q2, Q3 e Q4) que devem ser acionadas de forma sequencial e alternada, isso significa que é necessário desativar o interruptor previamente acionado para ativar o próximo interruptor na sequência configurada, alternando assim os acionamentos entre todos os interruptores. Dependendo da sequência de configuração dessas chaves, o motor gira em um sentido.

Quando ocorre o acionamento de uma das chaves, ela fecha o circuito e permite a passagem da corrente elétrica para o motor, fazendo com que ele gire em um sentido.

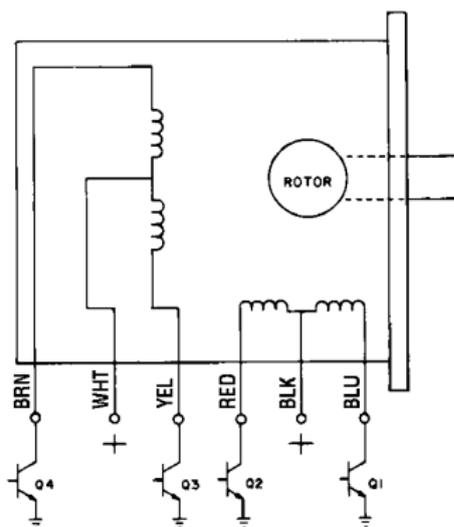


Figura 3.5: Diagrama Elétrico do Motor de Passo (THOMSON, 2023)

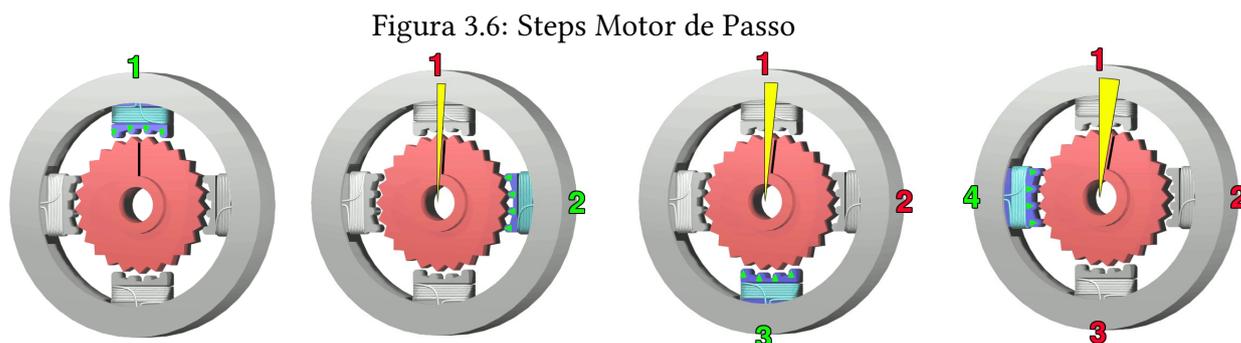


Figura 3.6: Steps Motor de Passo

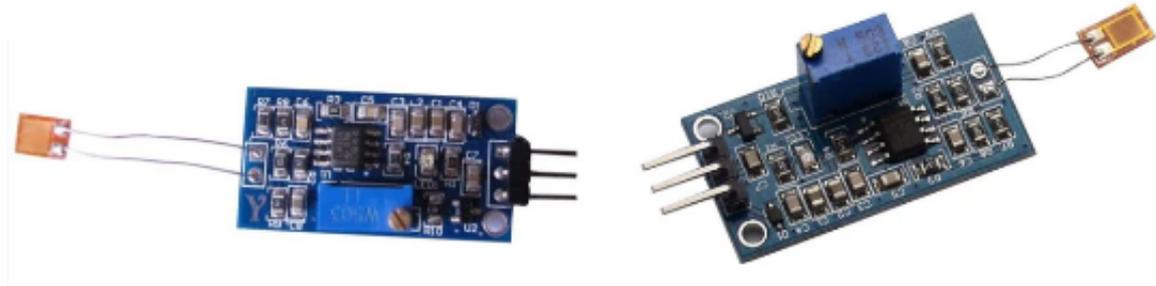
Fonte :(RIPKA; TIPEK, 2007)

Quando a próxima chave é acionada, a anterior é desativada, interrompendo a passagem da corrente e invertendo o sentido de rotação do motor como apresentado na Figura 3.6. Para a construção do circuito, pode ser utilizado qualquer tipo de componente que simule uma chave liga-desliga, como transistores.

### 3.1.4 Extensômetro

O Módulo Extensômetro *Sensor Strain Gauge* mede a deformação do corpo onde ele está instalado. Essa deformação é refletida em resistência, que pode ser lida e interpretada por um outro dispositivo e conseqüentemente, atuar em alguma ação. Sempre que o corpo ao qual o strain gauge está instalado for esticado, comprimido ou submetido a deformações, a resistência do sensor irá variar(ENSUS, 2023). A Figura 3.7 mostra esse módulo.

Figura 3.7: Extensômetro



Fonte: (SCIENTIFIC, 2023)

A variação de resistência do *strain gauge* pode ser muito pequena por isso é recomendável a utilização de uma ponte de Wheastone para geração de uma tensão mensurável no caso de uma variação de resistência.

Adicionalmente, o Módulo Extensômetro *Sensor Strain Gauge* possui circuito que gera uma tensão de saída proporcional a deformação sofrida pelo corpo, essa tensão já está amplificada para ser lida por um microcontrolador(SCIENTIFIC, 2023).

As especificações do extensômetro, de acordo com o fabricante se encontram na Tabela 3.4

Tabela 3.4: Ficha Técnica Extensômetro (ENSUS, 2023)

Modelo	BF350-3AA/1.5AA
Fator de sensibilidade	2
Limite de tensão	2%
Resistência	350Ω
Tensão de Operação	5V
Tipo de Saída	Analógica
Saída analógica de tensão	0-3.5V
Dimensões	32X17mm

### 3.1.5 Buzzer

Um buzzer é um dispositivo eletrônico que produz um som contínuo ou intermitente quando energizado. Geralmente, é composto por um elemento vibratório, como uma membrana ou um diafragma, que é acionado por uma corrente elétrica. Quando a corrente passa pelo buzzer, ele vibra, gerando ondas sonoras que percebemos como som. Esses dispositivos são comumente

usados em alarmes, sinalizações e em outros contextos onde é necessário um aviso sonoro. A Figura 3.8 mostra o buzzer.

Figura 3.8: Buzzer



Fonte: (PR, 2023)

Quando a corrente é aplicada ao buzzer, ela faz com que o disco de cerâmica se contraia ou se expanda. Alterar isso faz com que o disco ao redor vibre. Esse é o som que você ouve. Ao mudar a frequência da campainha, a velocidade das vibrações muda, o que altera o tom do som resultante (PR, 2023).

A Tabela 3.5 mostra a ficha técnica do buzzer, sendo algumas características:

- Buzzer Ativo;
- Emissão de sinais sonoros em frequência única em torno de 2khz;
- Ativado por corrente DC;
- Compatível com Arduino, AVR, PIC, AMR, Raspberry PI, etc.

Tabela 3.5: Ficha Técnica Buzzer (PR, 2023)

Tensão de trabalho	2.5 a 8 V
Tensão recomendada	5 V
Corrente máxima	40 mA
Diâmetro	12mm
Altura	10mm

### 3.1.6 Display LCD

O Display LCD (Figura 3.9) tem a função de responder a um comando do usuário com caracteres alfanuméricos. Os modelos podem ser com *Backlight (Back)* ou sem *Backlight*, que é a luz de fundo do display.

Figura 3.9: Display LCD



Fonte: (ELETROGATE, 2023)

#### Especificações Display LCD 16x4 com Back Verde

Um display perfeito para qualquer aplicação, altamente versátil por trabalhar com comunicação por 4 e 8bits e possuir um padrão de pinos que facilita a montagem. Possui como diferencial um maior número de linhas (4) sem a necessidade de vários CS(Chip Select) e uma proporção agradável, mostrando 64 caracteres. Pode ser usado em todas as aplicações, desde hobistas para comunicação com microcontroladores, para comunicação direta com o PC para CaseMod e também para aplicações profissionais(ELETROGATE, 2023).

Algumas características Display LCD 16x4 com Back Verde são(ELETROGATE, 2023):

- Tensão de Alimentação: 5V;
- Corrente: 3mA;
- N° de Colunas: 16;
- N° de Linhas: 4;
- Cor de Fundo: Verde;
- Cor de Escrita: Preto;
- Dimensões: 87mm(W) x 60mm(H) x max 13,5mm (LED *Backlight*).

### 3.1.7 Relé

Os relés são componentes eletromecânicos capazes de controlar circuitos externos de grandes correntes a partir de pequenas correntes ou tensões, ou seja, acionando um relé com uma pilha podemos controlar um motor que esteja ligado em 110 ou 220 volts, por exemplo. A Figura 3.10 mostra o relé utilizado neste projeto.

Figura 3.10: Relé



Fonte: (MASTERWALKERSHOP, 2023)

O Módulo Relé 5V 1 Canal permite que a partir de uma plataforma microcontrolada seja possível controlar cargas AC (alternada) de forma simples e prática. Por ter apenas 1 canal, é possível controlar apenas uma carga AC de até 10A. Comumente é utilizado em projetos de automação residencial para controle de lâmpadas, ventiladores e outras saídas que possam ser acionadas através de relé. A ficha técnica do relé está contida na Tabela 3.6.

Tabela 3.6: Ficha Técnica Relé (MASTERWALKERSHOP, 2023)

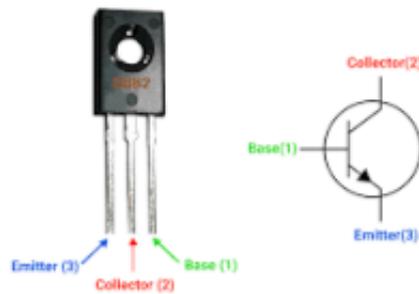
Capacidade do relé	30VDC/10A e 250VAC/10A
Tensão de Operação	3,3V – 5VDC
Corrente de operação	15 - 20mA
Tempo de resposta	5 - 10ms
LED indicador para presença de tensão	Sim
LED indicador para acionamento do relé	Sim

### 3.1.8 Transistor

Um transistor é um dispositivo semicondutor utilizado para amplificar ou modular sinais elétricos, além de desempenhar funções chave em circuitos eletrônicos. Ele é composto por

três camadas de material semicondutor (geralmente silício) e possui três terminais: emissor, base e coletor. A corrente elétrica ou voltagem aplicada ao terminal de base controla a corrente que flui entre o emissor e o coletor, permitindo assim a amplificação de sinais elétricos.

Figura 3.11: Transistor



Fonte: (COMPONENTS101, 2023)

O D882 é um transistor NPN de média potência que apresenta alta tensão de ruptura coletor-emissor e capacidade de transporte de corrente média (COMPONENTS101, 2023).

### 3.1.9 Sensor de Som / Detecção de Ruídos

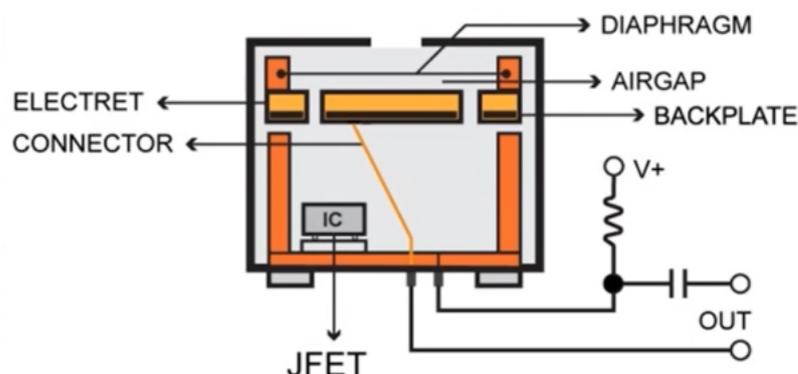
Este sensor possibilita a detecção de sons, a partir da variação de sua saída digital. Possui um microfone de condensador elétrico e pode ser usado em sistemas de alarme, por exemplo. A Tabela 3.7 apresenta a ficha técnica do microfone de eletreto.

Tabela 3.7: Ficha Técnica Microfone de Eletreto (ELETROGATE, 2023)

Sinal Limpo com boa forma de onda e corrente de saída	> 15 mA
Tensão de Operação	3.3V - 5V
Formato sinal de saída	Digital (0 -1)
Comparador	LM393
Possui furos para fácil instalação e fixação ao projeto	Sim

O limite de detecção pode ser ajustado através do potenciômetro presente no sensor que regulará a que nível de sinal sonoro a saída digital D0 vai alterar seu estado. Importante: a inversão na alimentação pode ocasionar a queima do módulo. A Figura 3.12 mostra o circuito do funcionamento do microfone de eletreto.

Figura 3.12: Circuito Microfone de eletreto



Fonte: (ELETROGATE, 2023)

Os Microfones de Eletreto (também conhecidos como microfones de condensador) são sensores de grande sensibilidade e tamanho compacto, sendo o tipo de microfone com um dos maiores leques de aplicações tanto profissionais quanto para projetos estudantis e hobbistas. Estes sensores são polarizados, trabalham com tensão entre 2 e 3V com consumo de 1mA e saída entre 9 e 20mV.

### 3.1.10 Registro de Pressão

Registro de Pressão é um dispositivo hidráulico que permite controlar o fluxo de água, desde a liberação ao bloqueio total da passagem de fluxo. Para o projeto, utilizou-se o registro "BASE REG PRESSAO MVS 3/4 DN20-B", apresentado na Figura 3.13 e a Figura 3.14 mostra o desenho técnico do registro, o qual é destinado tanto para fins comerciais quanto residenciais. Conforme informações obtidas no site da Deca, o custo desse registro é de R\$ 52,16. Essas informações foram acessadas em 17 de junho de 2023 DECA (2023).

As especificações do registro, de acordo com a ficha técnica DECA (2023), se encontram Tabela 3.8

### 3.1.11 Drive

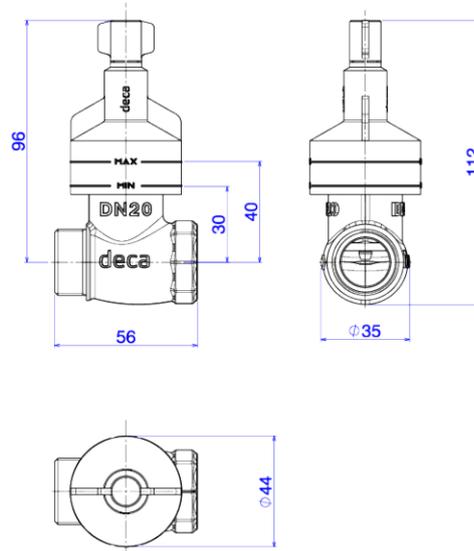
O Drive é um componente eletrônico projetado para controlar o funcionamento do motor. Seu uso se faz necessário para o controle adequado de um motor de passo, pois o mesmo é responsável por fornecer a corrente necessária e coordenar a sequência de energização das bobinas, permitindo assim, a geração de campos magnéticos que impulsionam o movimento do motor.

Figura 3.13: Figura Ilustrativa Registro de Pressão



Fonte: (DECA, 2023)

Figura 3.14: Desenho Técnico Registro de Pressão



Fonte: (DECA, 2023).

Neste projeto, o drive foi construído com transistores, diodos e resistências conforme Layout do drive construído no software Eagle Figura 3.15. O desenvolvimento do drive em uma placa de fibra de vidro será explicado na Seção 3.2.5.

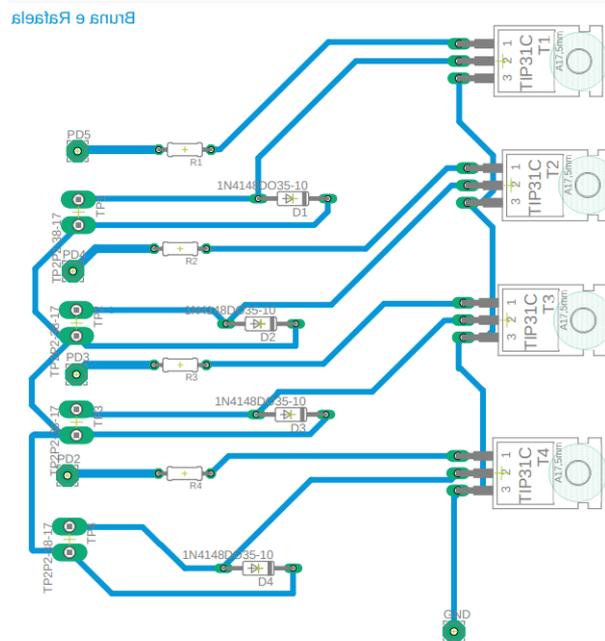
Os transistores desempenham um papel fundamental no controle do motor, pois são responsáveis por fornecer a corrente necessária para acionar as bobinas do motor de passo. Eles atuam como chaves eletrônicas, permitindo o fluxo de corrente através das bobinas de acordo com os sinais enviados.

Para escolha do transistor, foi necessário considerar as especificações do motor. Dado que o motor utilizado, possui uma tensão de operação de 12 V, e a resistência por bobina é  $21\Omega$ . Utilizando a Lei de Ohm na Equação 3.1, pode-se calcular a corrente que passa por cada bobina, aproximadamente 0,57A. Considerando essas informações, o transistor escolhido foi o TIP31C,

Tabela 3.8: Ficha Técnica Registro de Pressão DECA (DECA, 2023)

Linha	Base Registro De Pressão
Dimensões(AxLxC)	112mm X 56mm X 44mm
Material	Liga de Cobre (bronze e latão), Plásticos de Engenharia, Elastômeros
Peso líquido	0.303
Peso bruto	0.314
Número Norma / Decreto	NBR15704-1
Pressão mín/max	2 - 40 MCA
Bitola de entrada de água	3/4- DN20

Figura 3.15: Layout do circuito do Drive no software Eagle



Fonte: De autoria própria e(TAKITANI, 2023)

pois de acordo com o **tip31c-datasheet**, datasheet do dispositivo, possui uma capacidade de corrente nominal de até 3A e é capaz de suportar uma tensão de até 100V entre o coletor e o emissor. Portanto, como o motor de passo requer uma tensão de operação de 12V, o TIP31C é adequado para operar com a tensão de 12V necessária para o motor de passo.

$$V = R \cdot I \quad (3.1)$$

Um outro dispositivo utilizado no desenvolvimento do drive é o diodo. Ele é um dispositivo eletrônico semicondutor que possui dois terminais e é amplamente utilizado como um retificador de corrente elétrica. Ele permite que a corrente flua em apenas um sentido, bloqueando o fluxo reverso de corrente.

O diodo é utilizado no circuito de acionamento para assegurar a proteção do circuito contra tensões inversas prejudiciais. Ele atua como uma chave de acionamento automático, quando o diodo está na região de polarização direta (chaveamento fechado), permite que a corrente flua em apenas um sentido. Nessa condição, a resistência do diodo deve ser praticamente zero, se comportando como um circuito fechado. Já na região de polarização reversa (chaveamento aberto), o diodo bloqueia a passagem de corrente de maneira eficiente. Nessa situação, sua resistência é muito alta, se comportando como em um circuito aberto.(boylestad2013).

É necessário que o diodo esteja em paralelo com as chaves do motor para proteger o circuito contra tensões inversas geradas pelo armazenamento de energia nos indutores.

O diodo é conectado em paralelo de forma que o cátodo fique conectado à Tensão de Alimentação (VCC). Dessa forma, a energia armazenada no indutor é descarregada pelo diodo.

Foram utilizados quatro diodos do tipo 1N4148 no projeto. De acordo com as especificações do datasheet **won-top\_1n4148**, esses diodos possuem uma Tensão Reversa de Pico não repetitiva (VRM) de 100V e uma Tensão Reversa Média (VRRM) de 75V. Esses valores são superiores à tensão de operação do motor de passo, que é de 12V, garantindo assim uma margem de segurança adequada contra tensões reversas prejudiciais.

### 3.1.12 Sensor de Temperatura

Foi utilizada uma versão à prova d'água do sensor de temperatura DS18B20. Indicado para aplicações onde você precisa medir temperatura a longa distância e em ambientes úmidos. Como este é um sensor digital, a leitura do sensor não sofre interferências em função da distância. O DS18B20 possui resolução de 9 a 12 bits (configuráveis) para leituras de temperatura através de uma interface 1-Wire, de modo que apenas um fio (e terra) deve ser ligado ao microcontrolador.

Como cada DS18B20 contém um número de série único, vários DS18B20S podem ser conectados a um mesmo barramento 1-Wire. Isto permite a instalação e leitura de sensores de temperatura em diversos lugares diferentes. Este recurso é muito utilizado em controles

HVAC de climatização de ambientes, controle de temperatura no interior de edifícios, equipamentos ou máquinas, e monitoramento e controle de processos. A Figura 3.16 mostra esse sensor de temperatura.

Figura 3.16: Sensor de temperatura



Fonte: (MAKERHERO, 2023)

Importante: A temperatura máxima de resposta deste sensor é 125°C, mas como seu cabo é revestido em PVC, a sugestão é que se utilize o sensor em aplicações abaixo de 100°C.

A Tabela 3.9 apresenta a ficha técnica do sensor de temperatura utilizado.

Tabela 3.9: Ficha Técnica Sensor de Temperatura

Tensão de Alimentação	3.0 VDC a 5.5 VDC
Precisão	0.5°C de -10°C a + 85°C
Leitura de Temperatura	-55° a +125°C
Resolução	9 ou 12 bits
ID Único	64bits
Vários sensores podem ser utilizados no mesmo pino digital	
Alarme	limite de temperatura
Tempo de atualização	Menor que 750ms
Comprimento do cabo	90cm

### 3.1.13 Sensor de Fluxo / Sensor de Vazão

O sensor de Fluxo (Figura 3.17) é um dispositivo utilizado para monitorar e medir o fluxo de determinado fluido. Ele é projetado para detectar e quantificar o movimento do fluido em termos de volume, velocidade em um determinado período de tempo.

O sensor de Fluxo de Água 1/2 YF-S201B, utilizado no projeto, consiste em rotor, um ímã posicionado no eixo do motor e sensor de efeito Hall. A medida que o fluido passa pelo sensor de fluxo, o rotor é acionado e começa a girar. Esse movimento faz com que o campo magnético do ímã próximo ao sensor de efeito Hall mude periodicamente à medida que o polo sul e o polo norte passam por ele (**Robocoresensorfluxo**). A Tabela 3.10 mostra as características do sensor de fluxo.

Figura 3.17: Sensor de Fluxo de Água 1/2 YF-S201B



Fonte:(BYTEFLOP, 2023)

### 3.1.14 Sensor de nível de água

O sensor de nível de água para utilização em caixas d'água, tanques, reservatórios e outros recipientes.

O sensor de nível funciona como uma chave liga-desliga que pode acionar chaves, bombas, lâmpadas ou enviar um sinal para o microcontrolador como o Arduino, Pic ou Raspberry Pi. Pode ser utilizado tanto na parte superior do recipiente como na parte inferior, já que possui um anel de vedação (o-ring) e travamento por porca. O sensor deve ser instalado na posição vertical. A Figura 3.18 mostra o sensor de nível de água utilizado.

Tabela 3.10: Especificação Sensor de Fluxo de Água 1/2 YF-S201B (BYTEFLOP, 2023).

Modelo	YSF201
Tipo de sensor	Efeito Hall
Tensão de operação	5 - 24V
Corrente máxima	15mA
Faixa de Fluxo	1 - 30L/min
Pressão máxima	2,0 MPa
Frequência	7,5 * Fluxo(L/min)
Temperatura de trabalho	-25 a 80°C
Exatidão	10%

Figura 3.18: Sensor de nível



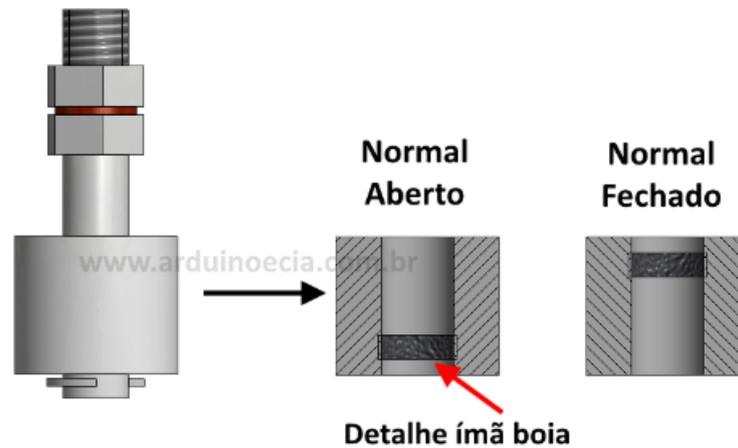
Fonte: (DIGITAL, 2023)

Este é um sensor feito de material plástico (PP, segundo o fabricante), composto de uma haste na qual desliza um cilindro feito de material flutuante. Esse cilindro possui um ímã que aciona um sensor magnético no meio da haste. Esta haste por sua vez fecha o contato dos 2 fios que saem do sensor.

A Figura 3.19 mostra o funcionamento do sensor de nível. O cilindro possui internamente apenas uma pequena parte preenchida com o ímã, o que torna possível invertermos o cilindro. Assim podemos ter um contato normal aberto ou normal fechado, adaptando o sensor às nossas necessidades:

Sozinho, pode, por exemplo, ser usado como uma chave que é acionada quando um reservatório está cheio, colocando o sensor na parte de cima do reservatório, ou vazio,

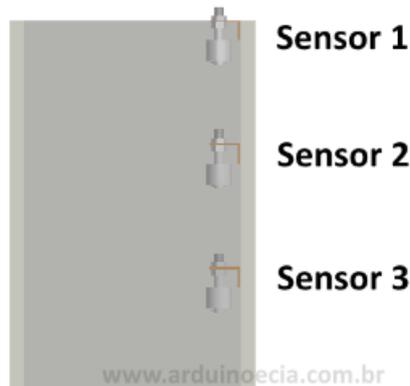
Figura 3.19: Sensor de nível



Fonte: (DIGITAL, 2023)

colocando o sensor na parte de baixo. Com mais sensores, é possível controlar vários níveis em um tanque, indicando ao Arduino se o meu tanque está cheio, se está pela metade ou se está (quase) vazio, conforme ilustrado na Figura 3.20. No projeto utilizamos o sensor para

Figura 3.20: Sensor de nível



Fonte: (DIGITAL, 2023)

apontar que há vazamento no tanque sendo assim utilizamos o sensor na parte de baixo.

### 3.1.15 Sensor de turbidez

O sensor de turbidez detecta a qualidade da água medindo o nível de turbidez. Ele é capaz de detectar partículas suspensas na água medindo a transmitância da luz e a taxa de espalhamento, que muda com a quantidade de sólidos suspensos totais (SST) na água. À medida que o TTS aumenta, o nível de turbidez do líquido aumenta (HPEROBOTICA, 2023).

Os sensores de turbidez podem ser usados na medição da qualidade da água em rios e riachos, medições de águas residuais e efluentes, pesquisa de transporte de sedimentos e medições laboratoriais (HPEROBOTICA, 2023).

A Figura 3.21 mostra o sensor de turbidez e a Tabela 3.11 apresenta suas especificações.

Figura 3.21: Sensor de turbidez



Fonte: (HPEROBOTICA, 2023)

Tabela 3.11: Especificações sensor de turbidez (HPEROBOTICA, 2023)

Tensão	3.3 - 5V (com circuito anti-reverso)
Saída	Simulação padrão (saída de nível alto e baixo pode ser definida)
Faixa de medição	0 - 1000 NTU

## 3.2 Métodos

Nesta seção, serão apresentadas as etapas metodológicas adotadas para a realização deste trabalho.

### 3.2.1 Protótipo do Sistema Hidráulico

Para a construção do protótipo foram utilizados dois recipientes de plástico que juntos simulam o ambiente real do transporte da água de um recipiente a outro.

Com a disposição da placa Arduino ligar o display LCD separadamente da ESP8266 para que os dados dos sensores fossem também mostrados em tempo real no LCD para verificar a

vazão, temperatura e demais sensores. Com a ESP8266 integrada ao projeto é possível conectar tais dados em uma página web para que possa ficar acessível de forma remota. Descrevemos a seguir os passos realizados para a criação do protótipo. Este protótipo usa um sensor de fluxo de água para medir a quantidade de água que passa pelo sistema, está localizado entre os dois recipientes de plástico. Com a ajuda de bombas de água colocadas nos dois recipientes de plásticos tem-se um sistema que simula o ambiente fechado e contínuo de um sistema de água (ou qualquer outro efluente).

A partir do sistema fechado criado, os demais sensores foram acrescentados: sensor de temperatura, sensor de turbidez, sensor extensômetro no cano que leva a um dos baldes (fazendo a monitoração da deformação no cano), sensor de som e sensor de nível, que é acionado ao ocorrer a vazão ou atingir o nível programado (não esperado). Dependendo da necessidade do projeto, as múltiplas aplicações de diversos setores dentro da sociedade (para o monitoramento de efluentes e possíveis riscos de vazamentos destes).

Os materiais necessários para o desenvolvimento do protótipo foram:

- Arduino Uno;
- Sensor de fluxo de água (YF-S201);
- Tubo de água;
- Baldes ou recipientes para coletar a água;
- Fios de conexão;
- Protoboard e jumpers;
- Fonte de alimentação para o Arduino (bateria ou adaptador de energia);
- Sensor de nível;
- Sensor de som;
- Buzzer;
- Sensor de turbidez;
- Sensor de temperatura;
- Sensor Extensômetro;

- Potenciômetro;
- Display LCD.

Os passos para a montagem foram:

- Conectamos o Sensor de Fluxo de Água ao Arduino:

O sensor de fluxo de água normalmente possui três pinos: VCC, GND e OUT. Conecte VCC ao 5V no Arduino, GND ao GND e OUT a um pino digital, como o pino 2.

- Conectamos dos Baldes ou Recipientes:

Posicionamos os baldes ou recipientes sob o sensor de fluxo de água para coletar a água que passa pelo sensor.

- Acrescentamos os demais sensores:

Realizando um bom posicionamento dos sensores para que possam coletar os devidos dados, conseguimos organizá-los no protótipo de forma organizada, para coletarem da melhor forma os possíveis dados do fluxo de água do sistema criado.

- Configuramos o Código no Arduino:

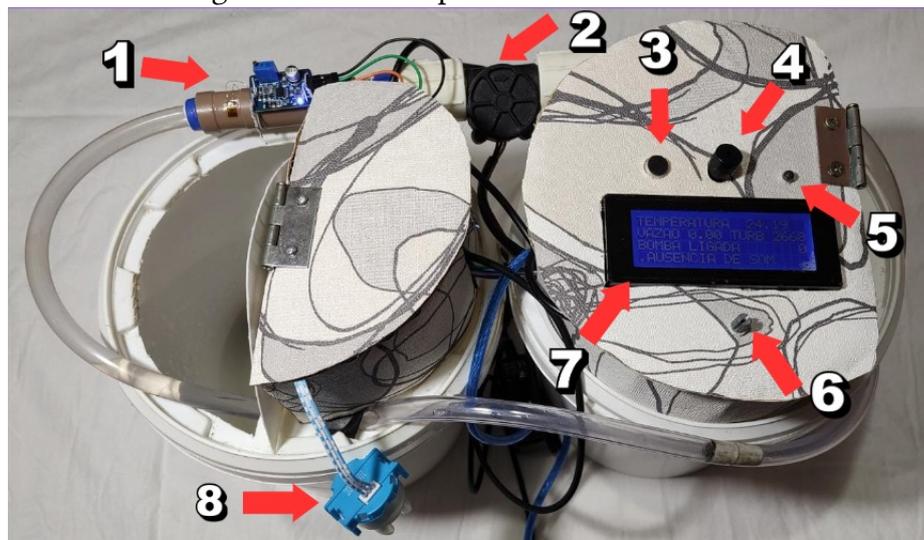
Utilizamos a IDE do Arduino para escrever o código. Para medir o tempo que o sensor está ativo e calcular a vazão com base nesse tempo deixamos este em aberto para verificar o fluxo máximo que passa por este e conseguir coletar possíveis alterações no sistema que levem a uma ruptura e seja necessário acionar a válvula (prevenindo possíveis vazamentos).

- Alimentação e Teste:

Podemos alimentar o arduino usando uma fonte de alimentação ou o próprio computador. Carregamos o código desenvolvido no Arduino e observamos a leitura da vazão na porta serial. Certificamos que o sensor de fluxo de água está instalado corretamente e que a água está fluindo através dele.

A Figura 3.22 mostra o protótipo do sistema hidráulico montado.

Figura 3.22: Protótipo do sistema hidráulico



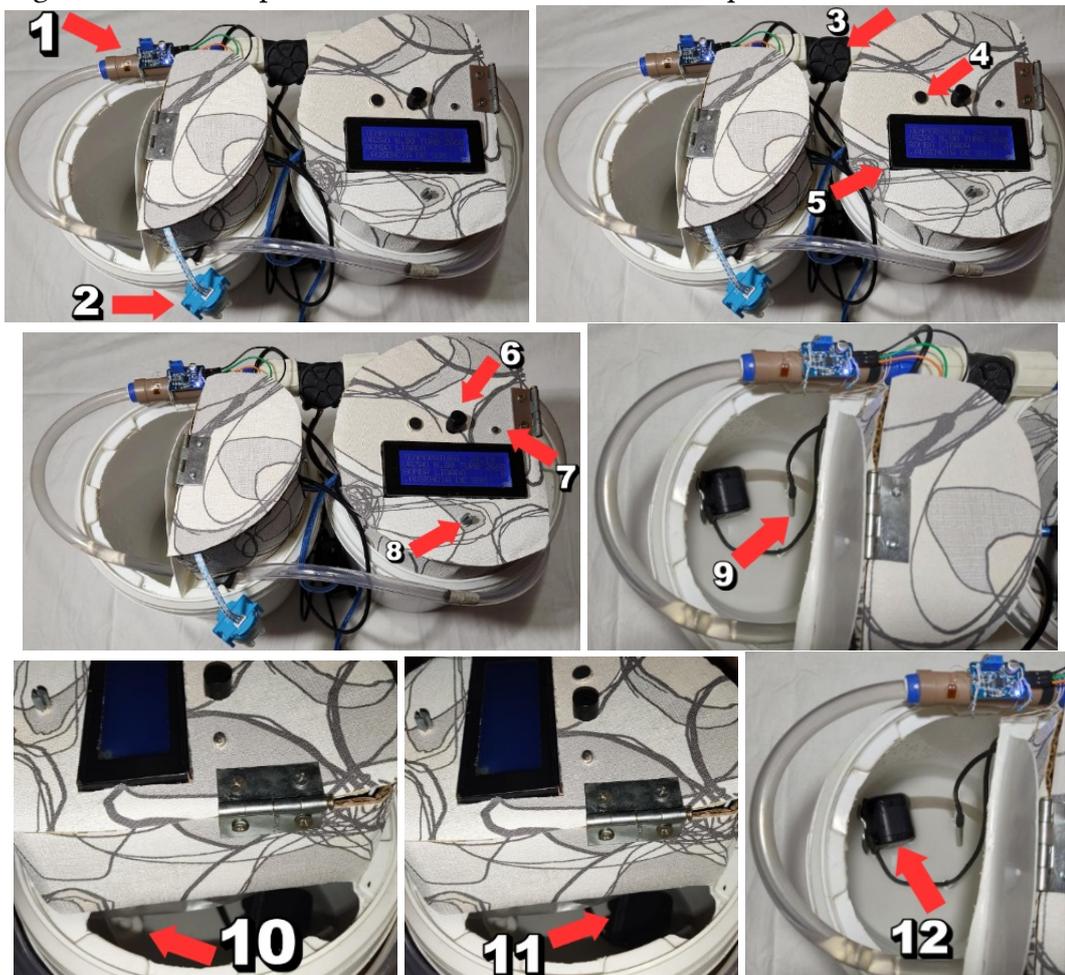
Fonte: De autoria própria

Os testes e a montagem do protótipo foram realizados no Laboratório de Telecomunicações da Unicamp FT. A Figura 3.23 ilustra o protótipo montado com todos os componentes necessários citados anteriormente.

Nas figuras é possível observar em 1: strain gauge, 2: sensor de turbidez, 3: sensor de vazão, 4: sensor de som, 5: display LCD, 6: buzzer; 7: led, 8: potenciômetro para controle da luz do display LCD, 9: sensor de temperatura, 10: sensor de nível, 11: 1ª bomba d'água e 12: 2ª bomba d'água.

O ciclo desenvolvido possibilita que a água passe por cada sensor e desenvolva (colete) a ação e o impacto que este desenvolve no ambiente e ou material, gerando um dado que é disposto de forma remota em uma página Web por meio de transmissão com o Wi-Fi (internet). É importante demonstrar e averiguar que temos à disposição um display LCD, que disponibiliza em tempo real dados da temperatura, vazão, turbidez, se a 1ª bomba d'água está ligada ou desligada e presença de som no ambiente.

Figura 3.23: Protótipo do sistema hidráulico com destaque aos sensores utilizados



Fonte: De autoria própria

### 3.2.2 Página web

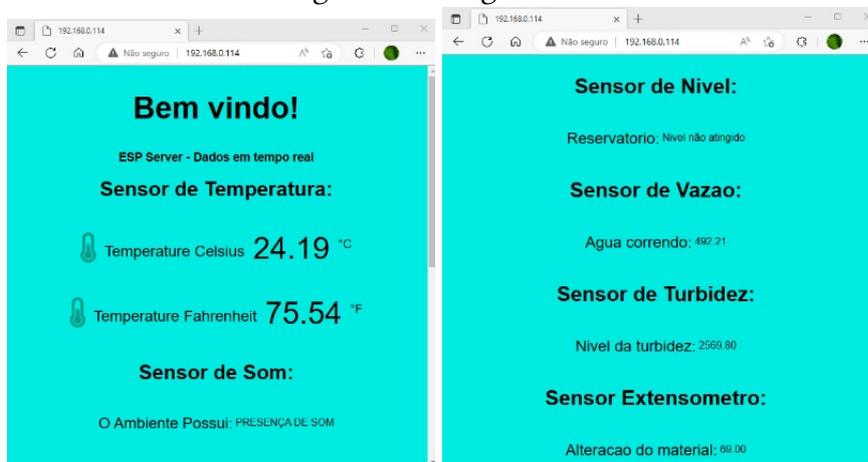
A linguagem utilizada para a construção da página foi o HTML (Linguagem de Marcação de Hipertexto), que é a linguagem padrão para a criação e o design de páginas web. Foi desenvolvido para descrever a estrutura básica de uma página web, usando uma sintaxe composta por elementos marcados por tags.

A estrutura fundamental do HTML consiste em elementos HTML que são representados por tags. Cada tag começa com um sinal de menor («") e termina com um sinal de maior (»"). As tags geralmente são usadas em pares, com uma tag de abertura e uma tag de fechamento, que é semelhante à tag de abertura, mas inclui uma barra ("/") antes do nome da tag.

A estrutura básica de uma página HTML é frequentemente composta por uma estrutura como a que se encontra na página criada e que pode ser verificado no código disponibilizado

no apêndice. A Figura 4.7 mostra a página desenvolvida, mostrando os dados coletados em tempo real dos sensores.

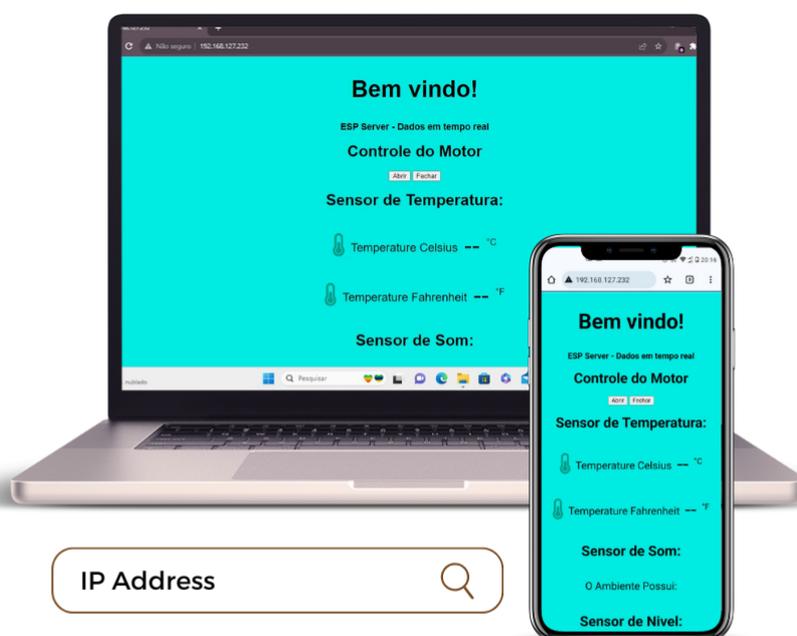
Figura 3.24: Página web



Fonte: De autoria própria

Para acessar a página web basta inserir o número do IP Address no navegador padrão, escolha do usuário, e já será carregado a página, conforme a Figura 3.25. Pode ser utilizado em qualquer dispositivo conectado a internet, como por exemplo notebooks, celulares e tablets.

Figura 3.25: Exemplo de tipos de utilização página web



Fonte: De autoria própria

### 3.2.3 Conexão Válvula e Motor de Passo

Para realizar a conexão entre o registro Deca 3/4 e o motor de passo, foram realizadas algumas adaptações para garantir um acoplamento eficiente. Essas adaptações se fizeram necessárias devido à diferença de construção e compatibilidade entre as duas peças, o que impossibilitava um encaixe perfeito sem intervenções adicionais.

Foi confeccionada uma bucha de aço com o objetivo de solucionar de acoplamento entre o motor e o registro proporcionando uma ligação sólida que não interferisse na rotação do motor, como indicada em vermelho Figura 3.26.

Figura 3.26: Acoplamento da bucha com o registro e o motor



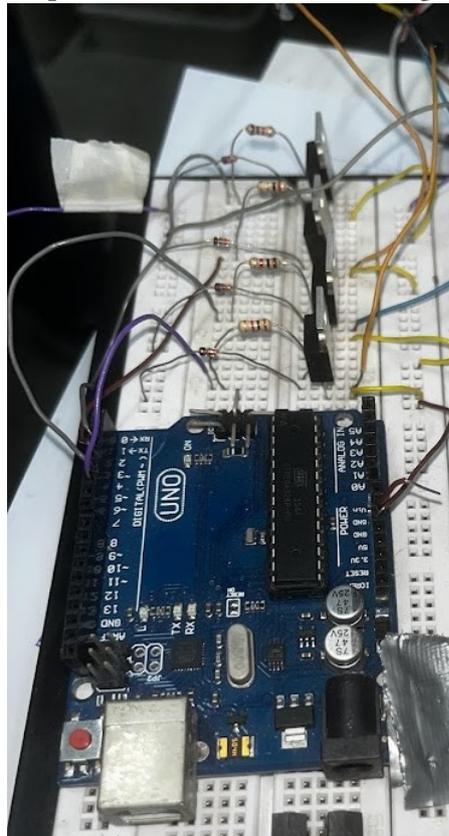
Fonte: De autoria própria em conjunto com (TAKITANI, 2023)

### 3.2.4 Montagem e código dos Circuitos do Sistema

Inicialmente, a montagem do sistema do motor foi feita utilizando a placa Arduino e o driver diretamente conectados em uma placa protoboard. A Figura 3.27 mostra essa montagem.

Na placa protoboard, os transistores foram devidamente conectados e interligados com as bobinas do motor de passo, garantindo o acionamento correto de cada uma delas. O Arduino foi

Figura 3.27: Acoplamento da bucha com o registro e o motor



Fonte: De autoria própria em conjunto com (TAKITANI, 2023)

programado para enviar os sinais de controle para os transistores, determinando a sequência e a direção do movimento do motor.

Para realizar os primeiros testes, foi desenvolvido um código em Arduino para verificar o funcionamento do acoplamento entre um motor e um registro. Esse código simula o movimento de abertura e fechamento do registro, permitindo avaliar se o acoplamento entre os componentes está funcionando corretamente.

Nesse trecho de código para Arduino, estamos definindo as variáveis `motorPin2`, `motorPin3`, `motorPin4` e `motorPin5` para representar os pinos de controle do motor de passo conectado ao Arduino. A atribuição `int i = 0;` é feita para inicializar a variável `i` com o valor zero. Essa variável é utilizada posteriormente no loop `for` para controlar o número de iterações.

```
int motorPin2 = 2; // Pino de controle do motor de passo 2
int motorPin3 = 3; // Pino de controle do motor de passo 3
int motorPin4 = 4; // Pino de controle do motor de passo 4
int motorPin5 = 5; // Pino de controle do motor de passo 5
```

```
int i = 0; // Inicialização da variável i com o valor zero
```

No contexto do Arduino, a função `setup()` é executada uma única vez quando o microcontrolador é ligado ou reiniciado. É nessa função que define a configuração inicial do seu programa, como a definição dos modos dos pinos utilizados. Neste trecho do código, estão sendo definidos que os pinos `motorPin2`, `motorPin3`, `motorPin4` e `motorPin5` são saídas, ou seja, serão utilizados para enviar sinais de controle para o motor de passo.

```
void setup()
{
    // Configuração dos pinos como saída
    pinMode(motorPin2, OUTPUT); // Pino de controle do motor de passo 2
    pinMode(motorPin3, OUTPUT); // Pino de controle do motor de passo 3
    pinMode(motorPin4, OUTPUT); // Pino de controle do motor de passo 4
    pinMode(motorPin5, OUTPUT); // Pino de controle do motor de passo 5
}
```

A função `loop()` é executada continuamente após a execução da função `setup()`. Nesse caso, a função "loop for" se repete 170 vezes. Dentro desse loop, está sendo acionando sequencialmente os pinos de controle do motor de passo para girar o motor em uma direção específica. Nesse caso, inicialmente o motor gira para o fechamento da registro, e posteriormente para a abertura do registro. A função `digitalWrite()` está sendo utilizada para definir o estado dos pinos, seja como HIGH (ligado) ou LOW (desligado), e a função `delay()` para introduzir uma pausa de 20 milissegundos entre cada acionamento do pino.

```
void loop()
{
    // Primeiro loop: acionando os motores de passo em sequência
    for (i = 0; i <= 170; i++)
    {
        digitalWrite(motorPin2, HIGH); // Aciona o motor de passo 2
        delay(20);
        digitalWrite(motorPin2, LOW); // Desativa o motor de passo 2
        digitalWrite(motorPin3, HIGH); // Aciona o motor de passo 3
    }
}
```

```
    delay(20);
    digitalWrite(motorPin3, LOW);    // Desativa o motor de passo 3
    digitalWrite(motorPin4, HIGH);   // Aciona o motor de passo 4
    delay(20);
    digitalWrite(motorPin4, LOW);    // Desativa o motor de passo 4
    digitalWrite(motorPin5, HIGH);   // Aciona o motor de passo 5
    delay(20);
    digitalWrite(motorPin5, LOW);    // Desativa o motor de passo 5
}
delay(5000); // Aguarda 5 segundos
// Segundo loop: acionando os motores de passo em sequência invertida
for (i = 0; i <= 170; i++)
{
    digitalWrite(motorPin5, HIGH);   // Aciona o motor de passo 5
    delay(20);
    digitalWrite(motorPin5, LOW);    // Desativa o motor de passo 5
    digitalWrite(motorPin4, HIGH);   // Aciona o motor de passo 4
    delay(20);
    digitalWrite(motorPin4, LOW);    // Desativa o motor de passo 4
    digitalWrite(motorPin3, HIGH);   // Aciona o motor de passo 3
    delay(20);
    digitalWrite(motorPin3, LOW);    // Desativa o motor de passo 3
    digitalWrite(motorPin2, HIGH);   // Aciona o motor de passo 2
    delay(20);
    digitalWrite(motorPin2, LOW);    // Desativa o motor de passo 2
}
delay(10000000); // Aguarda um tempo longo antes de repetir o loop
}
```

Após a primeira sequência de acionamentos, há um atraso de 5 segundos utilizando a função `delay(5000)`. Em seguida, é realizada uma segunda sequência de acionamentos, mas na direção oposta. Essa sequência de acionamentos repetitivos permite que o motor de passo

execute um movimento de rotação em uma direção por um determinado número de passos (no caso, 170) e depois retorne à posição inicial.

Após o primeiro teste, foi constatado que o motor de passo é capaz de controlar efetivamente a abertura e o fechamento total do registro. Com base nesse resultado, deu-se início à montagem do conjunto motor e drive em uma placa de fibra de vidro, conhecida como, Printed Circuit Board (PCB). Essa modificação foi realizada com o objetivo de proporcionar um transporte mais seguro e eficiente durante os testes realizado no Laboratório de Hidráulica, em Campinas, evitando o risco de desconexão dos fios das protoboards ou outras falhas na conexão.

Essa abordagem visa garantir a integridade do sistema durante o deslocamento e garantir que o conjunto motor e drive esteja pronto para ser utilizado nos testes futuros.

### 3.2.5 Desenvolvimento Placa PCB

Para o desenvolvimento de uma placa de circuito impresso (PCB), foram seguidos os seguintes passos e que estão ilustrados na Figura 3.28:

Figura 3.28: Procedimentos para o desenvolvimento da placa PCB: Layout Circuito no Eagle; Prensa Térmica; Placa com circuito.



Fonte: De autoria própria em conjunto com (TAKITANI, 2023)

- Projeto no software de design Eagle: Inicialmente, o drive, circuito montado da protoboard foi projetado em um software de design EAGLE, onde foram definidas as trilhas e componentes necessários.

- Layout do circuito: Após o projeto do circuito, foi realizado o layout, que envolve posicionar os componentes na placa, rotear as trilhas de conexão.

- Impressão do layout: Após a verificação do circuito, o layout finalizado foi impresso em um papel sulfite utilizando uma impressora comum.

- Preparação da placa: Foi utilizado uma placa de cobre revestida com objetivo de ser transferida o circuito impresso. A placa utilizada é feita de fibra de vidro.

- Transferência do layout para a placa: O layout impresso foi colocado sobre a placa, alinhado corretamente, e então foi aplicado pressão e calor utilizando uma prensa térmica. Isso permite que o layout impresso seja transferido para a placa por meio de um processo de transferência térmica.

- Correção: Após a revelação do circuito na placa, foi realizados retoques com caneta preta permanente para correções nas trilhas. Em seguida, a placa foi limpa com bombril para remover os resíduos.

- Perfurações dos orifícios para encaixe dos componentes: Para finalizar a PCB, foram realizados processos de perfuração dos orifícios para os componentes com o auxílio de uma máquina de perfuração.

- Corrosão da placa: Após o processo de perfuração dos orifícios, a placa foi submetida a um líquido corrosivo, o Percloroeto de Ferro, para remover o cobre do circuito. A placa foi mergulhada no Percloroeto de Ferro cerca de 30 minutos. Durante esse tempo, o líquido corrosivo reagiu com o cobre exposto, dissolvendo-o e criando o percurso das trilhas do circuito impresso. O cobre protegido pelo circuito, ou seja, as trilhas transferidas com o papel impresso, não são afetado pelo líquido corrosivo. Após o tempo necessário, a placa é removida do líquido corrosivo e lavada em água para interromper o processo de corrosão.

Após esses passos, a placa pôde receber os componentes eletrônicos e foi possível realizar as soldas e a montagem do circuito. A Figura 3.29 mostra algumas imagens do desenvolvimento da PCB.

Todos os materiais utilizadas foram disponibilizados pelo próprio laboratório da FT Unicamp.

### **3.2.6 Testes Realizados em Sistemas Hidráulicos**

Com base nos testes realizados foi possível concluir que o sistema construído atende aos requisitos necessários para abrir e fechar uma válvula em uma tubulação de água com uma pressão de aproximadamente 10 Metros de Coluna D'Água (MCA).

Figura 3.29: Procedimentos para o desenvolvimento da placa PCB: Layout Circuito no Eagle; Prensa Térmica; Placa com circuito.



Fonte: De autoria própria em conjunto com (TAKITANI, 2023)

Durante os testes, foi observado que, quando a válvula foi fechada, não houve passagem de água pela tubulação. Isso indica que a válvula foi fechada completamente, bloqueando o fluxo de água de forma eficiente.

Esses resultados são significativos, pois demonstram que o sistema de acoplamento entre o motor e a válvula foi projetado e implementado com sucesso, permitindo o controle preciso da abertura e fechamento da válvula, mesmo sob pressão de água.

As seções 3.2.3, 3.2.4 e 3.2.5 foram reproduzidos conforme o Trabalho de Conclusão de Curso Automatização de Válvula Integrada ao Aplicativo Telegram (TAKITANI, 2023), ao qual estamos dando continuidade.

## Capítulo 4

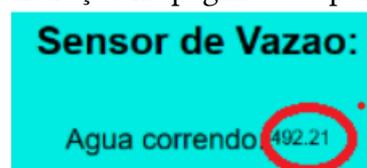
# Resultados e Discussões

Nesse capítulo, apresentam-se os principais resultados do projeto a respeito do protótipo de um sistema hidráulico para detecção de possíveis vazamentos em tubulações com válvula automatizada.

Em relação às medições e leituras, o sistema criado fechado coleta dados e os demonstra em tempo real. Ao acionar o sensor de nível o sistema para, indicando que a válvula deve ser fechada e demonstrando alguma irregularidade do sistema, ou seja, nesse caso em específico que o nível de água excedeu o nível esperado, por se tratar de uma lógica de vazão de água, o mesmo pode ser programado para a lógica invertida para a falta de água. Podendo ser aplicado em barragens de contenção de água, na indústria alimentícia detectando possíveis excessos ou faltas que seja necessário acionar uma válvula em si e tudo via rede de internet por intermediação da análise de uma página web que detém dados mostrando os mesmos em tempo real.

A Figura 4.1 mostra um exemplo de resultado para o sensor vazão medido em l/min:

Figura 4.1: Demonstração na página web para sensor de vazão

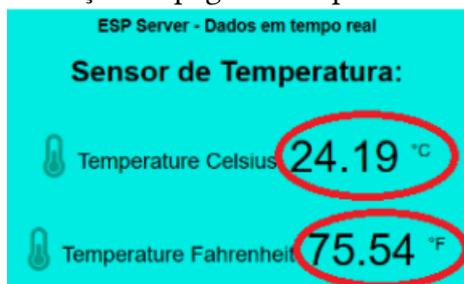


Fonte : De autoria própria

A Figura 4.2 mostra um resultado para o sensor de temperatura temos a medição em Celsius e Fahrenheit:

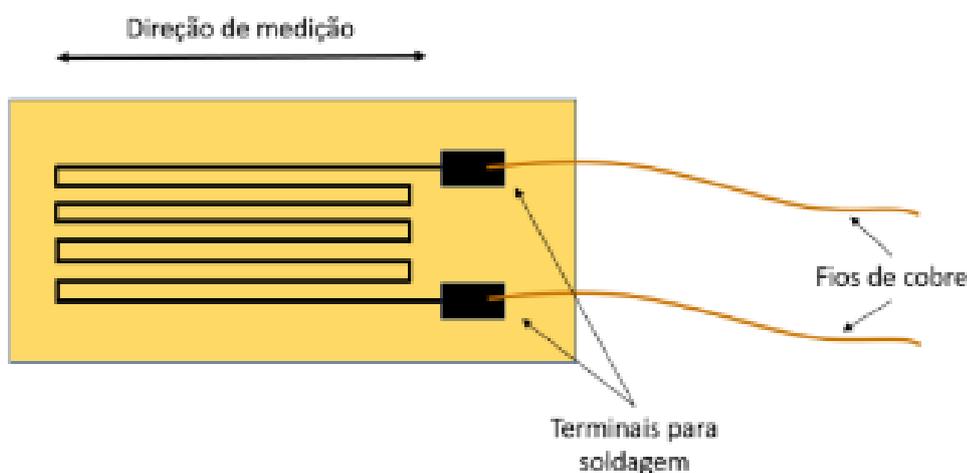
Para o sensor extensômetro temos uma medição de deformidade, conforme Figura 4.3.

Figura 4.2: Demonstração na página web para sensor de temperatura



Fonte : De autoria própria

Figura 4.3: Composição do sensor extensômetro

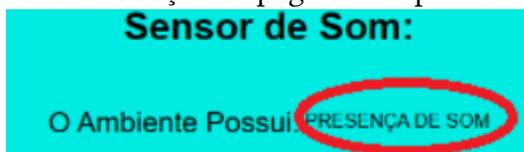


Fonte: (SISTEMAS ELETRÔNICOS - USP, 2023)

A alteração da resistência elétrica do extensômetro é justificada pela segunda Lei de Ohm, que afirma que a resistência elétrica de um condutor é proporcional ao comprimento do condutor. Com base nessa teoria, os extensômetros foram construídos de forma que ao serem tracionados, seu comprimento aumenta, e por consequência sua resistência também. As deformações costumam ser muito pequenas, e necessariamente, não podem ultrapassar o limite elástico do material, na prática ficam bem distantes do limite escoamento, pois os extensômetros são sensíveis, e podem ser danificados quando submetidos a grandes deformações.

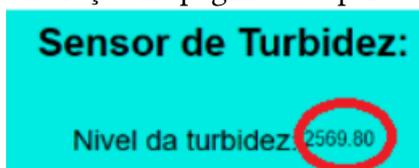
As Figuras 4.4, 4.5 e 4.6 mostram os resultados do sensor de som, de turbidez e de nível, respectivamente.

Figura 4.4: Demonstração na página web para sensor de som



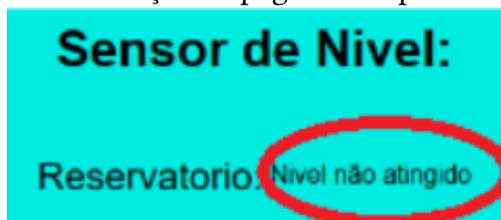
Fonte : De autoria própria

Figura 4.5: Demonstração na página web para sensor de turbidez



Fonte : De autoria própria

Figura 4.6: Demonstração na página web para sensor de nível



Fonte : De autoria própria

## 4.1 Conectividade e Transmissão de Dados

Para que os dados fossem transmitidos para a página web, foram seguidos os seguintes passos:

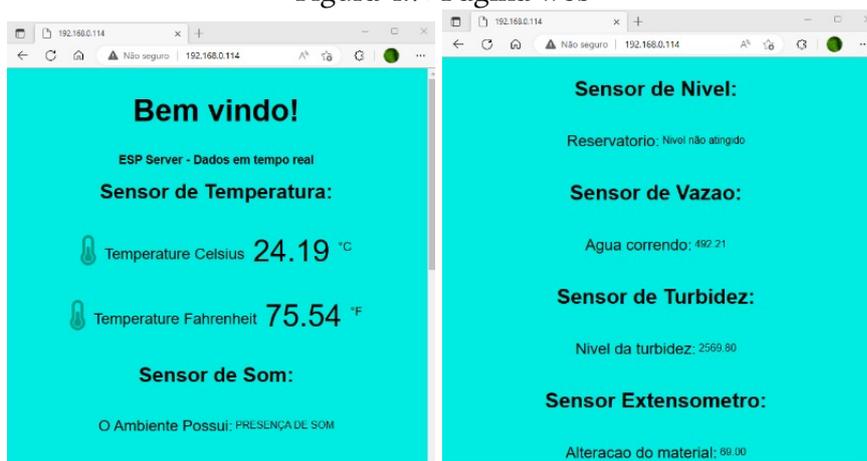
Primeiro quando utilizamos o arduino com um módulo ESP8266 para hospedar uma página web na própria placa a configuração do ESP8266:

- Programamos o ESP8266 usando o Arduino IDE.
- Definimos as configurações de rede SSID, senha, Servidor Web Embutido sendo parâmetros relevantes e importantes para que a página funcione. Descrevemos o código para criar um servidor web embutido no ESP8266. Bibliotecas como ESP8266WebServer são comumente usadas para simplificar essa tarefa.
- Criamos o conteúdo da página web em HTML e CSS Este conteúdo é incorporado diretamente no código do programa que roda no ESP8266.
- Tratamento de Requisições: O código do servidor web no ESP8266 é configurado para lidar com diferentes tipos de requisições HTTP.
- Quando um cliente (por exemplo, um navegador) faz uma requisição, o código do servidor processa essa requisição.

- Resposta para o Cliente: O servidor envia de volta a resposta apropriada para o cliente. Isso pode incluir o código HTML da página, arquivos CSS, imagens ou qualquer outro recurso solicitado. Comunicação HTTP: A comunicação entre o cliente (navegador) e o servidor ESP8266 é realizada usando o protocolo HTTP.
- Dependendo das necessidades, podemos usar HTTP ou HTTPS para a transmissão de dados.
- Interatividades: A página web inclui formulários e elementos interativos, o ESP8266 pode receber dados do cliente através de requisições POST ou GET. que vemos acontecendo em tempo real, conseguindo os dados obtidos. como podemos visualizar nas imagens abaixo:

A Figura 4.7 mostra a página desenvolvida, mostrando os dados coletados em tempo real dos sensores.

Figura 4.7: Página web



Fonte : De autoria própria

A transmissão ocorreu conforme esperado não houve problemas de conectividade que impedissem de apresentar os dados coletados na página. A conexão Wi-Fi foi estabelecida com sucesso e não houve interrupções no processo de transmissão.

A página web foi desenvolvida para mostrar os dados coletados em tempo real dos sensores. A interface da página web avalia a clareza e a acessibilidade das informações. A interface web foi projetada de forma a facilitar a interpretação dos dados pelos usuários, com elementos visuais intuitivos (segundo com a descrição do sensor e o resultado de sua leitura).

Testes de funcionalidade foram realizados, o protótipo passou por testes onde a leitura da vazão foi observada na porta serial, indicando que houve uma verificação preliminar do funcionamento, foram identificados e corrigidos problemas durante a construção como

alteração de biblioteca, para conectividade com o sensor específico, identificação do sensor mediante a uma porta digital ou analógica.

A válvula pode ser acionada de duas formas, através da página web e pelo acionamento do sensor de nível.

Durante a interação com o sensor de nível, foi possível observar no monitor serial, conforme mostrada na Figura 4.8, que a conexão com o Wi-Fi já havia sido estabelecida, e quando acionado a mensagem recebida pelo monitor serial é que o nível foi atingido e logo em seguida que a mensagem que está ocorrendo o fechamento da válvula.

Figura 4.8: Monitor Serial

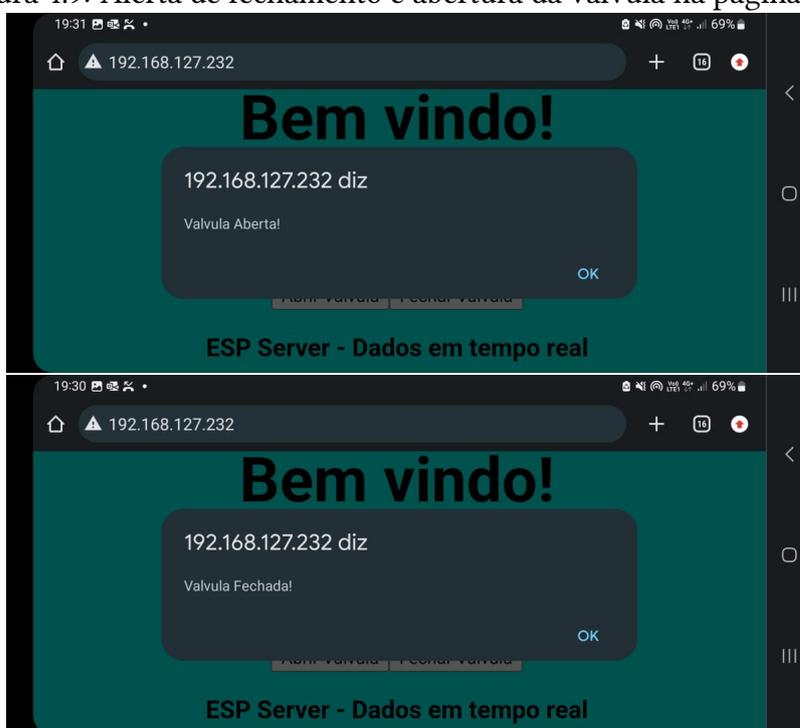
```
-----  
19:19:51.033 -> Connecting to WiFi...  
19:19:52.015 -> Connecting to WiFi...  
19:19:52.015 -> Connected to WiFi  
19:19:52.015 -> 192.168.127.232  
19:20:05.021 -> Nível atingido  
19:20:19.787 -> Nível atingido. Fechando motor.
```

Fonte : De autoria própria

Ao ser acionado o botão fechar ou abrir na página web. Sincronamente, o motor de passo começou a girar a válvula para ser fechada ou aberta. Na página web é exibido um alerta com o texto "Válvula Aberta!"ou "Válvula Fechada!"foi exibida para informar o usuário sobre o processo de abrir ou fechar a válvula foi encerrado,conforme mostrada na Figura 4.9

Uma vez que o protótipo foi desenvolvido para monitorar o fluxo de água e outros parâmetros, utilizando sensores como sensor de fluxo de água, sensor de temperatura, sensor extensômetro, sensor de som, sensor de turbidez e sensor de nível. A avaliação da utilidade depende da precisão na coleta de dados, da capacidade de detecção de anomalias e da eficácia na transmissão desses dados para a página web, que se mostrou eficaz. Além disso, a aplicabilidade do protótipo depende das necessidades específicas do projeto em que pode ser aplicado na indústria(tubulações), rede de esgoto ou em meio a uma área florestal, monitoramento de barragens e etc. Sendo ente bem modularizado para distintas aplicações.

Figura 4.9: Alerta de fechamento e abertura da válvula na página web



Fonte : De autoria própria

# Capítulo 5

## Conclusões

A concepção e implementação do protótipo de sistema hidráulico representam um passo significativo em direção à inovação na gestão de tubulações. A capacidade de detectar rupturas de forma precisa e, mais ainda, corrigi-las por meio de uma válvula automatizada, introduz uma solução prática e eficaz para desafios comuns no contexto hidráulico.

A inclusão de uma página web para visualização amplia ainda mais a utilidade do sistema, proporcionando uma interface intuitiva para monitorar e controlar as operações remotamente. Essa acessibilidade não apenas simplifica a gestão do sistema, mas também fortalece a capacidade de resposta a eventos inesperados, contribuindo para a eficiência operacional e a redução de desperdícios.

Ao integrar tecnologia e funcionalidade, este protótipo não apenas atende às demandas contemporâneas por inovação, mas também sinaliza um futuro promissor para a gestão inteligente de recursos hídricos. A detecção precoce de rupturas, a correção automatizada e a interface web intuitiva convergem para criar um sistema hidráulico avançado e adaptável, alinhado com os princípios de sustentabilidade e eficiência. Este trabalho não é apenas uma conquista individual, mas uma contribuição valiosa para o progresso no campo da automação e gestão inteligente de sistemas hidráulicos.

Portanto, a continuidade deste trabalho em direção à automação da válvula representa não apenas uma extensão natural do TCC anterior (TAKITANI, 2023), mas também um avanço significativo no que diz respeito à funcionalidade da válvula em si. Ao configurar a utilização da válvula para correção de vazamentos em rupturas nas tubulações, o projeto demonstrou, de maneira concreta, os benefícios tangíveis dessa abordagem. Essas vantagens, aliadas à gestão inteligente dos recursos hídricos, revelam o potencial expansivo da

automação da válvula, extrapolando os limites do projeto inicial. Portanto, este trabalho não apenas amplia as fronteiras do conhecimento estabelecido no TCC anterior (TAKITANI, 2023), mas também oferece uma solução prática e inovadora com aplicações promissoras em diversas áreas.

# Referências Bibliográficas

BANZI, M. **Getting Started with Arduino**. [S.l.]: O'Reilly Media, Inc., 2009. ISBN 9780596555108.

BRITES, F. G.; SANTOS, V. P. d. A. **Motor de Passo**. Niterói - RJ, jul. 2008.

BYTEFLOP. **Sensor de Fluxo de Água 1/2 YF-S201**. Disponível em: <<https://www.byteflop.com.br/sensor-de-fluxo-de-agua-12-yf-s201>>. Acesso em: 20 jun. 2023.

COMPONENTS101. **D882 Transistor**. Disponível em: <<https://components101.com/transistors/d882-transistor>>. Acesso em: 5 dez. 2023.

DECA. **Ficha técnica DECA BASE REG PRESSAO MVS 3/4 DN20-B**. Disponível em: <<https://www.deca.com.br/ambientes/acabamentos-e-instalacao/itens-de-instalacao/base-para-registro-de-pessao/Base-Registro-de-Presso-MVS-Base-3-4-4416-202>>. Acesso em: 14 jun. 2023.

DIGITAL, I. **Sensor de Nível de Água**. Disponível em: <<https://www.institutodigital.com.br/produto/sensor-de-nivel-de-agua/>>. Acesso em: 5 dez. 2023.

ELETROGATE. **Guia Completo do Display LCD Arduino**. Disponível em: <<https://blog.eletrogate.com/guia-completo-do-display-lcd-arduino/>>. Acesso em: 5 dez. 2023.

ELETROGATE. **Módulo Sensor de Som KY-037**. Disponível em: <<https://www.eletrogate.com/modulo-sensor-de-som-ky-037>>. Acesso em: 5 dez. 2023.

ELETRÔNICOS, C. C. C. **Placa DOIT ESP32 - ESP32-WROOM-32D - WiFi / Bluetooth**. 2016-2022. Disponível em: <<https://curtocircuito.com.br/placa-doit-esp32-esp32-wroom-32d-wifi-bluetooth.html>>. Acesso em: 17 jun. 2023.

EMBRAPA. **Agricultura irrigada e sua importância na produção de alimento**. 2022. Disponível em: <<https://www.embrapa.br/busca-de-noticias/-/noticia/71510678/artigo-agricultura-irrigada-e-sua-importancia-na-producao-de-alimento-nexo-agua-alimento>>.

ENSUS. **Extensometria: Strain Gauge - O que é e quando utilizar**. Disponível em: <<https://ensus.com.br/extensometria-strain-gauge-o-que-e-quando-utilizar/>>. Acesso em: 5 dez. 2023.

GLEICK, P. H. (Ed.). **Water in Crisis: A Guide to the World's Fresh Water Resources**. Oxford, New York e associated companies; Berlin, Ibadan: Oxford University Press, 1993. p.

Includes bibliographical references. ISBN 0-19-307627-3. Disponível em: <<http://www.oup-usa.org>>.

HELLER, L.; PÁDUA, V. L. d. P. **Abastecimento de água para consumo humano**. 2. ed. Belo Horizonte: Editora UFMG, 2010. v. 2. 2 v. P. (Ingenium). Inclui bibliografia. ISBN 978-85-7041-845-6.

HPERBOTICA. **Sensor de Turbidez**. Disponível em: <<https://www.hperobotica.com.br/arduino/sensores-modulos-e-shield/sensor-de-turbidez>>. Acesso em: 5 dez. 2023.

MAKERHERO. **Sensor de Temperatura DS18B20 à Prova D'Água**. Disponível em: <<https://www.makehero.com/produto/sensor-de-temperatura-ds18b20-a-prova-dagua/>>. Acesso em: 5 dez. 2023.

MASTERWALKERSHOP. **Como Usar com Arduino Módulo Relé 5V 1 Canal**. Disponível em: <<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-rele-5v-1-canal>>. Acesso em: 5 dez. 2023.

MINISTÉRIO DO DESENVOLVIMENTO REGIONAL SECRETARIA NACIONAL DE SANEAMENTO. **Diagnóstico Temático Serviços de Água e Esgoto: visão geral ano de referência 2020**. 2021. Disponível em: <[https://www.gov.br/mdr/pt-br/assuntos/saneamento/snis/produtos-do-snis/diagnosticos/DIAGNOSTICO\\_TEMATICO\\_VISAO\\_GERAL\\_AE\\_SNIS\\_2021.pdf](https://www.gov.br/mdr/pt-br/assuntos/saneamento/snis/produtos-do-snis/diagnosticos/DIAGNOSTICO_TEMATICO_VISAO_GERAL_AE_SNIS_2021.pdf)>. Acesso em: 2 jul. 2022.

MULTILÓGICA-SHOP. (EQUIPE). **Arduino Guia Iniciante**. Baseado no "Guia del Arduino" criado pela Tienda de Robótica da Colômbia. Disponível em: <[https://edisciplinas.usp.br/pluginfile.php/3252633/mod\\_resource/content/1/Guia\\_Arduino\\_Iniciante\\_Multilogica\\_Shop.pdf](https://edisciplinas.usp.br/pluginfile.php/3252633/mod_resource/content/1/Guia_Arduino_Iniciante_Multilogica_Shop.pdf)>. Acesso em: 1 jun. 2023.

MULTIPEÇAS. **Placa de Desenvolvimento NodeMCU Lua WiFi com ESP8266 ESP-12E V3.0**. Disponível em: <<https://multipecas.curitiba.br/produto/placa-de-desenvolvimento-nodemcu-lua-wifi-com-esp8266-esp-12e-v3-0/>>. Acesso em: 5 dez. 2023.

PR, E. D. **Aula 36 - O Buzzer Toca Música**. Disponível em: <[https://aluno.escoladigital.pr.gov.br/sites/alunos/arquivos\\_restritos/files/documento/2022-12/aula36\\_o\\_buzzer\\_toca\\_musica\\_m1.pdf](https://aluno.escoladigital.pr.gov.br/sites/alunos/arquivos_restritos/files/documento/2022-12/aula36_o_buzzer_toca_musica_m1.pdf)>. Acesso em: 5 dez. 2023.

RIPKA, P.; TIPEK, A. (Ed.). **Modern Sensors Handbook**. [S.l.]: ISTE Ltd, 2007. ISBN 978-1-905209-66-8.

SCHOOL, W. S. (Ed.). **How much water is there on, in, and above the Earth?** 2019. Disponível em: <<https://www.usgs.gov/special-topics/water-science-school/science/how-much-water-there-earth>>. Acesso em: 2 jul. 2022.

SCIENTIFIC, M. **What is a Strain Gauge**. Disponível em: <<https://www.michsci.com/what-is-a-strain-gauge/>>. Acesso em: 5 dez. 2023.

SILVEIRA, A. L. M. d. **Página do Arduino Uno**. Disponível em: <<http://www.um.pro.br/arduino/index.php?c=ArduinoUno>>. Acesso em: 5 dez. 2023.

SISTEMAS ELETRÔNICOS - USP, I. de. **Extensômetros Elétricos**. Disponível em: <[https://sistemas.eel.usp.br/docentes/arquivos/5840793/LOM3086/Extensometros\\_eletricos.pdf](https://sistemas.eel.usp.br/docentes/arquivos/5840793/LOM3086/Extensometros_eletricos.pdf)>. Acesso em: 5 dez. 2023.

TAKITANI, R. Y. d. S. Automatização de Válvula Integrada ao Aplicativo Telegram. **Tese de Conclusão de Curso**, 2023.

THOMSON, a. m. **Product Selection And Engineering Guide**. Disponível em: <<http://www.ece.ualberta.ca/~ee401/parts/data/AirpaxCatalog.pdf>>. Acesso em: 30 jun. 2023.

# Apêndice A

## Código Desenvolvido

```
// Import required libraries

#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Hash.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define bobina2 D1 //amarelo
#define bobina3 D2 //roxo
#define bobina4 D3 //azul
#define bobina5 D4 //verde
const int levelSensorPin = D5; // Pino de leitura do sensor de nível

// Data wire is connected to GPIO 4
#define ONE_WIRE_BUS 8

// Setup a oneWire instance to communicate with any OneWire devices
```

```
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature sensor
DallasTemperature sensors(&oneWire);

// Variables to store temperature values
String temperatureF = "";
String temperatureC = "";
String presencaSOM = "";
String Nivel = "";
String vazao= "";
String turbi= "";
String curvatura= "";
float vazao1=0;
;
bool motorLigado = false;

//double calc_NTU(double volt);
float NTU = 0;
double volt;
//-----sensor vazão-----
const int INTERRUPT_SENSOR = 0; //interrupt = 0 equivale ao pino digital 2
const int PINO_SENSOR = 2;
double contador;
float fluxoAcumulado=0;
float metroCubico=0;
float ContaAgua=0;
float calculoVazao;

//-----sensor vazão-----

//-----sensor de turbidez-----
```

```
//double calc_NTU(double volt);
//double NTU = 0.0;
//-----sensor de turbidez-----

// Timer variables
unsigned long lastTime = 0;
unsigned long timerDelay = 30000;

// Replace with your network credentials

const char* ssid = "xxxxxxxxxxxx";
const char* password = "xxxxxxxxxxxx";

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);
//-----extensometro-----
void fechar(){
for (int i=0;i<=170;i++)
{
digitalWrite(bobina2, HIGH); // Aciona o motor de passo 2
delay(20);
digitalWrite(bobina2, LOW); // Desativa o motor de passo 2
digitalWrite(bobina3, HIGH); // Aciona o motor de passo 3
delay(20);
digitalWrite(bobina3, LOW); // Desativa o motor de passo 3
digitalWrite(bobina4, HIGH); // Aciona o motor de passo 4
delay(20);
digitalWrite(bobina4, LOW); // Desativa o motor de passo 4
digitalWrite(bobina5, HIGH); // Aciona o motor de passo 5
delay(20);
digitalWrite(bobina5, LOW); // Desativa o motor de passo 5
```

```
}
delay(1000);
}
void abrir(){
for (int i=0;i<=170;i++)
{
digitalWrite(bobina5, HIGH); // Aciona o motor de passo 5
delay(20);
digitalWrite(bobina5, LOW); // Desativa o motor de passo 5
digitalWrite(bobina4, HIGH); // Aciona o motor de passo 4
delay(20);
digitalWrite(bobina4, LOW); // Desativa o motor de passo 4
digitalWrite(bobina3, HIGH); // Aciona o motor de passo 3
delay(20);
digitalWrite(bobina3, LOW); // Desativa o motor de passo 3
digitalWrite(bobina2, HIGH); // Aciona o motor de passo 2
delay(20);
digitalWrite(bobina2, LOW); // Desativa o motor de passo 2
}
delay(1000);
}
//-----extensometro-----
String lerExtensometro(){
float alteracao;
String curvatura;
alteracao = analogRead(A0); //avaliador do sensor
Serial.println(alteracao);
if(alteracao){
curvatura = "modificacao";
Serial.println("modificacao:");
}
}
```

```
else{
curvatura = "Sem modificacao";
Serial.println("sem modificacao:");
}
return String(alteracao);
}

//-----sensor de turbi-----
String lerturbi(){

int sensorValue = analogRead(A0);
float voltage = sensorValue * (5.0/ 1024.0);

NTU = -(1120.4*voltage*voltage)+(5742.3*voltage)-4352.9;

NTU = -1*NTU;

//NTU = 1232131;
//delay (741);

return String(NTU);
}

//-----sensor de vazao-----

String lervazao(){
```

```
float calculoVazao;
float sensorValue = analogRead(A0);
//attachInterrupt(0, 1, RISING); //Configura a porta digital 2, para interrup

contador = sensorValue;
//interrupts();
//delay(1000);
//noInterrupts();

calculoVazao = (contador * 2.25);
fluxoAcumulado = fluxoAcumulado + (calculoVazao / 1000);
metroCubico = fluxoAcumulado / 1000;
ContaAgua = metroCubico * 3.59;
calculoVazao = calculoVazao * 60;
calculoVazao = calculoVazao/ 1000;
vazao1 = calculoVazao + vazao1;

return String(vazao1);

}
//-----sensor de som-----
String lerPresencaSom(){
bool som;
String presencaSOM;
som = digitalRead(0)>0; //avaliador do sensor
Serial.println(som);
if(som){
presencaSOM = "PRESENÇA DE SOM";
Serial.println("PRESENCA DE SOM");
}
}
```

```
else{
presencaSOM = "AUSENCIA DE SOM";
Serial.println("AUSENCIA DE SOM");
}
return String(presencaSOM);
}
//-----sensor de nivel-----
String lerNivel(){
String Nivel;
int estado = digitalRead(levelSensorPin); //estado é igual a leitura digital
  //Serial.print("Estado sensor : "); /*Printa "Estado do sensor:" */
  // Serial.println(estado); //Printa a leitura de estado/
  delay(100); //atraso de 0,1s/
  if(estado){
    estado = 1;
    Nivel = "Nivel atingido";
    Serial.println("Nivel atingido");
  }
  else{
    estado = 0;
    Nivel = "Nivel não atingido";
    // Serial.println("Nivel não atingido");
  }
return String(Nivel);
}
//-----sensor de Temperatura-----
String readDSTemperatureC() {
  // Call sensors.requestTemperatures() to issue a global temperature and Re
  sensors.requestTemperatures();
  float tempC = sensors.getTempCByIndex(0);

  if(tempC == -127.00) {
```

```
    Serial.println("Failed to read from DS18B20 sensor");
    return "--";
} else {
    Serial.print("Temperature Celsius: ");
    Serial.println(tempC);
}
return String(tempC);
}
```

```
String readDSTemperatureF() {
    // Call sensors.requestTemperatures() to issue a global temperature and Re
    sensors.requestTemperatures();
    float tempF = sensors.getTempFByIndex(0);

    if(int(tempF) == -196){
        Serial.println("Failed to read from DS18B20 sensor");
        return "--";
    } else {
        Serial.print("Temperature Fahrenheit: ");
        Serial.println(tempF);
    }
    return String(tempF);
}
```

```
//-----pagina web-----
```

```
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/c
    <style>
        html {
```

```
    font-family: Arial;
    display: inline-block;
    margin: 0px auto;
    text-align: center;
}
h2 { font-size: 3.0rem; }
p { font-size: 3.0rem; }
.units { font-size: 1.2rem; }
.ds-labels{
    font-size: 1.5rem;
    vertical-align:middle;
    padding-bottom: 15px;
}
</style>
</head>
<body>
  <h2>Bem vindo!</h2>
  <h3>ESP Server - Dados em tempo real</h3>
  %CONTROL_FORMS%

  <h1>Sensor de Temperatura:</h1>
  <p>
    <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
    <span class="ds-labels">Temperature Celsius</span>
    <span id="temperaturec">%TEMPERATUREC%</span>
    <sup class="units">&deg;C</sup>
  </p>
  <p>
    <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
    <span class="ds-labels">Temperature Fahrenheit</span>
    <span id="temperaturef">%TEMPERATUREF%</span>
```

```
    <sup class="units">&deg;F</sup>
<h1>Sensor de Som:</h1>
</p>
    <span class="ds-labels">O Ambiente Possui:</span>
    <span id="presencaSOM">%PRESENCASOM%</span>
</p>
    <h1>Sensor de Nivel:</h1>
</p>
    <span class="ds-labels">Reservatorio:</span>
    <span id="Nivel">%NIVEL%</span>
</p>
<h1>Sensor de Vazao:</h1>
</p>
    <span class="ds-labels">Agua correndo:</span>
    <span id="vazao">%VAZAO%</span>
</p>
<h1>Sensor de Turbidez:</h1>
</p>
    <span class="ds-labels">Nivel da turbidez:</span>
    <span id="turbi">%TURBI%</span>
</p>
<h1>Sensor Extensometro:</h1>
</p>
    <span class="ds-labels">Alteracao do material:</span>
    <span id="curvatura">%CURVATURA%</span>
</p>
<body bgcolor="SlateBlue1">
</body>
<script>
setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
```

```
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("temperaturec").innerHTML = this.responseText;
        }
    };
    xmlhttp.open("GET", "/temperaturec", true);
    xmlhttp.send();
}, 10000) ;

setInterval(function ( ) {
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("temperaturef").innerHTML = this.responseText;
        }
    };
    xmlhttp.open("GET", "/temperaturef", true);
    xmlhttp.send();
}, 10000) ;

setInterval(function ( ) {
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("presencaSOM").innerHTML = this.responseText;
        }
    };
    xmlhttp.open("GET", "/presencaSOM", true);
    xmlhttp.send();
}, 10000) ;

setInterval(function ( ) {
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
```

```
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("Nivel").innerHTML = this.responseText;
        }
    };
    xmlhttp.open("GET", "/Nivel", true);
    xmlhttp.send();
}, 10000) ;

setInterval(function ( ) {
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("vazao").innerHTML = this.responseText;
        }
    };
    xmlhttp.open("GET", "/vazao", true);
    xmlhttp.send();
}, 10000) ;

setInterval(function ( ) {
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("turbi").innerHTML = this.responseText;
        }
    };
    xmlhttp.open("GET", "/turbi", true);
    xmlhttp.send();
}, 10000) ;

setInterval(function ( ) {
    var xmlhttp = new XMLHttpRequest();
```

```
xhttp.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200) {  
        document.getElementById("curvatura").innerHTML = this.responseText;  
    }  
};  
xhttp.open("GET", "/curvatura", true);  
xhttp.send();  
}, 10000) ;
```

```
</script>
```

```
</html>rawliteral";
```

```
// Replaces placeholder with DS18B20 values
```

```
String processor(const String& var){
```

```
    //Serial.println(var);
```

```
    if(var == "TEMPERATUREC"){
```

```
        return temperatureC;
```

```
    }
```

```
    else if(var == "TEMPERATUREF"){
```

```
        return temperatureF;
```

```
    }
```

```
    else if(var == "presencaSOM"){
```

```
        return presencaSOM;
```

```
    }
```

```
    else if(var == "Nivel"){
```

```
        return Nivel;
```

```
    }
```

```
    else if(var == "vazao"){
```

```
        return vazao;
```

```
    }
```

```
    else if(var == "turbi"){
```

```
    return turbi;
  }
  else if(var == "curvatura"){
    return curvatura;
  }
  else if(var == "CONTROL_FORMS"){
    // Include the motor control forms
    return R"rawliteral(
<h1>Controle do Motor</h1>
<button type='button' onclick='acionarMotor("abrir")'>Abrir Valvula</but
<button type='button' onclick='acionarMotor("fechar")'>Fechar Valvula</b
    <script>
function acionarMotor(acao) {
    fetch('/controlarMotor?acao=${acao}')
      .then(response => response.text())
      .then(data => alert(data))
      .catch(error => console.error('Erro:', error));
}

    setTimeout(function(){
      window.location.replace("/");
    }, 1000); // Espera 1 segundo antes de redirecionar
    </script>
)rawliteral";
  }
  return String();
}

void setup(){

  Serial.begin(115200);
  Serial.println();
```

```
// Start up the DS18B20 library
sensors.begin();

temperatureC = readDSTemperatureC();
temperatureF = readDSTemperatureF();
presencaSOM = lerPresencaSom();
Nivel= lerNivel();
vazao = lervazao();
turbi = lerturbi();
curvatura= lerExtensometro();

// Connect to Wi-Fi
WiFi.begin(ssid, password);
Serial.println("Connecting to WiFi");
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println();
// Print ESP Local IP Address
Serial.println(WiFi.localIP());

// Route for root / web page
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html, processor);
});

server.on("/temperaturec", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", temperatureC.c_str());
});
```

```
server.on("/temperaturef", HTTP_GET, [] (AsyncWebServerRequest *request) {
    request->send_P(200, "text/plain", temperatureF.c_str());
});
server.on("/presencaSOM", HTTP_GET, [] (AsyncWebServerRequest *request) {
    request->send_P(200, "text/plain", presencaSOM.c_str());
});
server.on("/Nivel", HTTP_GET, [] (AsyncWebServerRequest *request) {
    request->send_P(200, "text/plain", Nivel.c_str());
});
server.on("/vazao", HTTP_GET, [] (AsyncWebServerRequest *request) {
    request->send_P(200, "text/plain", vazao.c_str());
});
server.on("/turbi", HTTP_GET, [] (AsyncWebServerRequest *request) {
    request->send_P(200, "text/plain", turbi.c_str());
});
server.on("/curvatura", HTTP_GET, [] (AsyncWebServerRequest *request) {
    request->send_P(200, "text/plain", curvatura.c_str());
});
// ...

server.on("/controlarMotor", HTTP_GET, [] (AsyncWebServerRequest *request) {
    if (request->hasParam("acao")) {
        String acao = request->getParam("acao")->value();
        if (acao == "abrir") {
            abrir();
            alert('Valvula Aberta');
            motorLigado = true;

            request->send(200, "text/plain", "Valvula Aberta");
        } else if (acao == "fechar") {
            fechar();
            alert('Valvula Fechada');
        }
    }
});
```

```
        motorLigado = false;
        request->send(200, "text/plain", "Valvula Fechada");
    } else {
        request->send(400, "text/plain", "Acao invalida");
    }
} else {
    request->send(400, "text/plain", "Parametro 'acao' ausente");
}
});
```

```
// Start server
server.begin();

//-----som-----
pinMode(0, INPUT);
//-----som-----

//-----nivel-----
pinMode(levelSensorPin, INPUT);
//-----nivel-----

//-----vazao-----
pinMode(16, INPUT);

//delay(1000);

//-----vazao-----

//-----turbi-----
//pinMode(A0, INPUT);
//-----turbi-----

//-----extensometro-----
```

```
pinMode(A0, INPUT);
//-----extensometro-----

}

void loop(){

    if ((millis() - lastTime) > timerDelay){
        temperatureC = readDSTemperatureC();
        temperatureF = readDSTemperatureF();
        presencaSOM = lerPresencaSom();
        Nivel = lerNivel();
        vazao= lervazao();
        turbi = lerturbi();
        curvatura= lerExtensometro();
        lastTime = millis();
        contador ++;

    }

    String nivel = lerNivel();
    if (nivel == "Nível atingido") {
        fechar();
        Serial.println("Nível atingido. Fechando motor.");
        motorLigado = false;
    }
}
```