

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

JONATAS DA SILVA RODRIGUES ROCHA

**DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE SUPERVISÓRIO E
AQUISIÇÃO DE DADOS WEB PARA UMA PLANTA DE FILTROS
GRANULARES**

TOLEDO

2025

JONATAS DA SILVA RODRIGUES ROCHA

**DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE SUPERVISÓRIO E
AQUISIÇÃO DE DADOS WEB PARA UMA PLANTA DE FILTROS
GRANULARES**

**Development of a Web Data Acquisition and Supervisory Control System for
a Granular Filter Plant**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica do Curso de Bacharelado em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. José Dolores Vergara Dietrich

**TOLEDO
2025**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

JONATAS DA SILVA RODRIGUES ROCHA

**DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE SUPERVISÓRIO E
AQUISIÇÃO DE DADOS WEB PARA UMA PLANTA DE FILTROS
GRANULARES**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia Eletrônica
do Curso de Bacharelado em Engenharia
Eletrônica da Universidade Tecnológica Federal
do Paraná.

Data de aprovação: 02/Julho/2025

Prof. José Dolores Vergara Dietrich, Dr.
Universidade Tecnológica Federal do Paraná

Prof. Ednei Luiz Miotto, Dr.
Universidade Tecnológica Federal do Paraná

Prof. Evandro Marcos Kolling, Dr.
Universidade Tecnológica Federal do Paraná

**TOLEDO
2025**

AGRADECIMENTOS

Ao olhar para trás, percebo o quanto esta trajetória foi marcada por desafios, aprendizados e, acima de tudo, pelas pessoas incríveis que estiveram ao meu lado. Este trabalho é resultado não apenas do meu esforço, mas também do apoio e amor daqueles que acreditaram em mim e me inspiraram a seguir em frente.

Primeiramente, agradeço à minha parceira e esposa, Sara, que esteve comigo em cada momento dessa jornada. Juntos, enfrentamos os altos e baixos que vieram com a graduação e a vida de pais. Ao nosso filho, meu pequeno, que chegou no meio do caminho para encher nossas vidas de alegria e dar um significado ainda maior a esta conquista, dedico este trabalho como um símbolo do quanto sonhos podem ser realizados com determinação e amor.

Aos meus colegas de graduação, Caio Lorrán e João Vitor, meu muito obrigado pelas conversas, risadas e apoio durante esses anos. Vocês tornaram o percurso mais leve e inesquecível.

À minha mãe, Marta, e ao meu pai, Josias, que sempre acreditaram no meu potencial, mesmo quando eu duvidava de mim mesmo. Vocês me ensinaram o valor do esforço e da resiliência, e por isso, serei eternamente grato. À minha avó Noêmia, pelo amor incondicional e palavras de sabedoria que sempre aqueceram meu coração e me deram forças para continuar.

Agradeço também aos professores e colegas que, de alguma forma, contribuíram para minha formação, e à Secretaria do Curso, pelo suporte nas questões administrativas que fizeram diferença ao longo da graduação.

Por fim, este trabalho é para todos aqueles que acreditam no poder da educação e do esforço conjunto. A vocês, meu sincero muito obrigado!

Primeira Lei: Um robô não pode ferir um ser humano ou, por omissão, permitir que um ser humano sofra algum mal. Segunda Lei: Um robô deve obedecer as ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens contrariem a Primeira Lei. Terceira Lei: Um robô deve proteger sua própria existência desde que tal proteção não entre em conflito com a Primeira e Segunda Leis (ASIMOV, 1950).

RESUMO

Este trabalho acadêmico teve como objetivo desenvolver um sistema de Supervisão e Aquisição de Dados (SCADA) utilizando ferramentas de Internet das Coisas (IoT). O SCADA é uma tecnologia amplamente utilizada na automação industrial para o monitoramento e controle eficiente de processos. Por sua vez, a IoT permite a conexão e integração de dispositivos inteligentes, possibilitando a coleta e transmissão de dados em tempo real. A combinação dessas tecnologias visa otimizar a automação de um permeâmetro de filtros granulares, promovendo maior eficiência e precisão no processo. A implementação do SCADA permite a criação de uma interface gráfica intuitiva, na qual os operadores podem monitorar e controlar os dispositivos e processos de forma centralizada.

Palavras-chave: scada; iot; automação.

ABSTRACT

This academic work aims to develop a Supervisory Control and Data Acquisition (SCADA) system using Internet of Things (IoT) tools. SCADA is a technology widely used in industrial automation for the efficient monitoring and control of processes. IoT enables the connection and integration of smart devices, allowing real-time data collection and transmission. The combination of these technologies seeks to optimize the automation of a granular filter permeameter, promoting greater efficiency and accuracy in the process. The implementation of the SCADA system provides an intuitive graphical interface, through which operators can centrally monitor and control devices and processes.

Keywords: scada; iot; automation.

LISTA DE FIGURAS

Figura 1 – Estrutura da planta experimental do permeâmetro.	13
Figura 2 – Elementos de um sistema SCADA	15
Figura 3 – Princípios de comunicação	15
Figura 4 – Internet das Coisas	16
Figura 5 – Modelo OSI	17
Figura 6 – Pilha de protocolos e suas equivalências	18
Figura 7 – Ipv4 e Ipv6	19
Figura 8 – Comparação entre camada IoT e TCP/IP	20
Figura 9 – Barramento I ² C	21
Figura 10 – Resistores Pull-Ups	22
Figura 11 – Comunicação I2C	22
Figura 12 – Arduino Mega 2560	26
Figura 13 – Placa de prototipação ESP32	27
Figura 14 – Arduino IDE	28
Figura 15 – Interface do Bubble	29
Figura 16 – Interface do <i>Firestore Realtime Database</i>	30
Figura 17 – Desenho esquemático mostrando a topologia geral do projeto	31
Figura 18 – Código configuração comunicação I ² C	33
Figura 19 – Código configuração comunicação com <i>Firestore</i>	34
Figura 20 – Formato dos dados enviados JSON	34
Figura 21 – Tela de monitoramento dos dados	35
Figura 22 – Circuito individual tela de monitoramento	38
Figura 23 – Exemplo de dados armazenados no Google Sheets	38

LISTA DE SÍMBOLOS

ARP	Protocolo de Resolução de Endereços (<i>Address Resolution Protocol</i>)
DHCP	Protocolo de Configuração de Host Dinâmico (<i>Dynamic Host Configuration Protocol</i>)
DNS	Sistema de Nomes de Domínio (<i>Domain Name System</i>)
FTP	Protocolo de transferência de arquivos (<i>File transfer protocol</i>)
HTTP	Protocolo de Transferência de Hipertexto (<i>Hypertext Transfer Protocol</i>)
HTTPS	Protocolo de Transferência de Hipertexto Seguro (<i>Hypertext Transfer Protocol Secure</i>)
IHM	Interface Homem Máquina
IoT	Internet das Coisas (<i>Internet of Things</i>)
IP	Protocolo de Internet (<i>Internet Protocol</i>)
ISO	<i>International Organization for Standardization</i>
I ² C	<i>Inter-Integrated Circuit</i>
OSI	<i>Open System Interconnection</i>
RARP	Protocolo de Resolução Reversa de Endereços (<i>Reverse Address Resolution Protocol</i>)
R _p	Resistores <i>pull-ups</i>
SCADA	Sistemas de Controle Supervisório e Aquisição de Dados
SCL	<i>Serial Clock Line</i>
SDA	<i>Serial Data Line</i>
SMTP	Protocolo de Transferência de Correio Simples (<i>Simple Mail Transfer Protocol</i>)
SSL	<i>Secure Sockets Layers</i>
TCP	Protocolo de Controle de Transmissão (<i>Transmission Control Protocol</i>)
UDP	Protocolo de Datagrama do Usuário

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Considerações iniciais	10
1.2	Objetivos	11
1.2.1	Objetivos específicos	11
1.3	Justificativa	11
1.4	Estrutura do trabalho	12
2	REFERENCIAL TEÓRICO	13
2.1	Permeâmetro e Filtros Granulares	13
2.2	Sistemas Supervisórios	14
2.3	Comunicação	15
2.3.1	Internet da Coisas	16
2.3.2	Protocolo TCP/IP	17
<u>2.3.2.1</u>	<u>Camada de aplicação</u>	17
<u>2.3.2.2</u>	<u>Camada de transporte</u>	19
<u>2.3.2.3</u>	<u>Camada de internet</u>	19
<u>2.3.2.4</u>	<u>Camada física e enlace</u>	20
2.3.3	Arquitetura IoT	20
2.3.4	Protocolo I ² C	21
2.4	Central de Controle	22
2.4.1	Armazenamento de dados em nuvem	23
2.4.2	Ambiente Gráfico em Sistemas SCADA	23
3	MATERIAIS E MÉTODOS	25
3.1	Materiais	25
3.1.1	Microcontroladores	25
<u>3.1.1.1</u>	<u>Arduino Mega</u>	25
<u>3.1.1.2</u>	<u>ESP32</u>	26
3.1.2	<i>SOFTWARES</i>	27
<u>3.1.2.1</u>	<u>Arduino IDE</u>	27
<u>3.1.2.2</u>	<u>Bubble</u>	28
<u>3.1.2.3</u>	<u>Firestore</u>	28

3.2	Métodos	30
3.2.1	Arquitetura Geral do Sistema	30
3.2.2	Comunicação entre Arduino e ESP32	32
3.2.3	Transmissão de Dados à <i>Firebase</i>	32
3.2.4	Visualização dos Dados	33
3.2.5	Considerações Técnicas	35
4	RESULTADOS E DISCUSSÕES	37
4.1	Visualização e Interação com os Dados	37
4.2	Exportação de Dados e Armazenamento Local	38
4.3	Limitações e Possibilidades de Expansão	39
5	CONCLUSÃO	40
	REFERÊNCIAS	41

1 INTRODUÇÃO

Os sistemas de supervisão e controle de processos desempenham um papel fundamental na automação industrial. Combinando dispositivos e *softwares* especializados, esses sistemas permitem os operadores a monitorar e controlar de maneira eficiente os processos de produção. Essa abordagem é especialmente valiosa em ambientes de difícil acesso e na automatização de tarefas repetitivas, garantindo maior eficiência e redução de falhas (JUNIOR, 2019).

Entre as ferramentas mais utilizadas está o SCADA, que permite monitorar e controlar sistemas de automação por meio de interfaces gráficas. Suas principais funções incluem a geração de gráficos e relatórios, registro de eventos, detecção de sensores, envio remoto de parâmetros e comandos para equipamentos (CARVALHO, 2018).

O avanço da engenharia eletrônica e de *software* tem impulsionado significativamente o desenvolvimento de sistemas de automação. Ferramentas de modelagem, simulação e sistemas operacionais em tempo real também têm contribuído para essa evolução, integrando as áreas de computação e automação industrial (LUGLI; SANTOS, 2019). Equipamentos como os CLPs (Controladores Lógicos Programáveis) desempenham um papel histórico e crucial nesse contexto. Contudo, seu custo elevado inviabiliza algumas aplicações, abrindo espaço para dispositivos de menor custo, como Arduino e ESP32, que oferecem soluções práticas para automação de baixo custo (IDEALI, 2021). No entanto, é importante destacar que essas soluções não se aplicam a todos os contextos, especialmente em ambientes industriais que demandam maior rigor quanto à robustez e confiabilidade do *hardware* e *software*.

Entre as diversas possibilidades de aplicação desses dispositivos, uma delas é o uso em ensaios geotécnicos, como os de permeabilidade de solos granulares. Esse tipo de ensaio é realizado com o auxílio de um permeâmetro, equipamento utilizado para determinar a capacidade de um solo em permitir a passagem de água através de seus poros. O controle e a aquisição automática dos dados desse processo podem tornar o ensaio mais preciso e eficiente.

Este trabalho propõe o desenvolvimento de uma aplicação SCADA com interface gráfica em ambiente web, utilizando as placas Arduino e ESP32 para a coleta e monitoramento de dados em um sistema de análise de permeabilidade de solos granulares.

1.1 Considerações iniciais

Os sistemas SCADA têm se destacado pela capacidade de integrar tecnologia acessível com funcionalidades avançadas de supervisão e controle. Este trabalho visa abordar as oportunidades de aplicação de dispositivos como o ESP32 e Arduino em sistemas de baixo custo, destacando seu papel na automação de processos específicos, como a análise de permeabilidade de solos.

1.2 Objetivos

Projetar um sistema supervisorio para um permeâmetro já existente cuja interface seja web, de forma que possa ser acessada em qualquer dispositivo que se conecte à internet.

1.2.1 Objetivos específicos

- Garantir que o sistema apresente confiabilidade na leitura e escrita de dados.
- Configurar o banco de dados *Firebase* para armazenar as informações de forma segura.
- Projetar uma interface web, de baixo custo e embarcável, que possibilite a interação do usuário com o processo em tempo real.

1.3 Justificativa

Este projeto justifica-se pela necessidade de criar um sistema eficiente e de baixo custo para automação e controle supervisorio, a fim de monitorar a análise de permeabilidade de solos granulares, visando a otimização desse processo. A implementação desse sistema busca melhorar a eficiência, precisão e gestão do sistema, além de facilitar a detecção de falhas e a tomada de decisões mais rápidas e assertivas.

Existem diversas soluções disponíveis no mercado para automação e controle de sistemas, incluindo opções pagas e de código aberto. Neste projeto, optou-se por utilizar uma solução de baixo custo e de código aberto, que possa ser embarcada. O sistema proposto permitirá o registro e armazenamento automático e em tempo real dos dados de análise de permeabilidade de solos, possibilitando um acesso mais fácil às informações e análises, contribuindo assim para uma tomada de decisão mais ágil e eficiente.

Além disso, a implementação desse sistema de monitoramento eficiente e acessível atende às necessidades específicas da análise de permeabilidade de solos granulares. Essa análise desempenha um papel crucial em estudos geotécnicos e geológicos, sendo amplamente utilizada em projetos de engenharia civil, construção de estradas, fundações e obras de infraestrutura. A obtenção de dados precisos e confiáveis sobre a permeabilidade do solo é essencial para garantir a segurança e a estabilidade dessas estruturas.

Ao adotar uma abordagem supervisionada, o sistema proposto permitirá um controle mais rigoroso do processo de análise de permeabilidade de solos granulares. Isso inclui a capacidade de identificar prontamente quaisquer desvios nos resultados, possibilitando uma investigação imediata e correção, se necessário. Além disso, a análise em tempo real dos dados coletados fornecerá uma visão detalhada do comportamento do solo, auxiliando na identificação de padrões e tendências relevantes.

A escolha de uma solução de baixo custo e de código aberto oferece vantagens adicionais, tornando o sistema mais acessível a instituições de pesquisa, laboratórios e empresas de menor porte, que podem se beneficiar desse sistema de monitoramento sem comprometer seus recursos financeiros. Além disso, a natureza de código aberto oferece flexibilidade e a possibilidade de personalização para atender às necessidades específicas da planta de análise de permeabilidade de solos.

Em suma, a implementação desse sistema de monitoramento apresenta uma forma de monitorar e permitir a otimização dos processos de análise de solos, fornecendo uma alternativa econômica em relação aos sistemas tradicionais de automação e controle supervísório.

Dado o contexto dos sistemas supervísórios e da aplicação de sistemas SCADA, é importante destacar que o equipamento permeâmetro — utilizado para medir a permeabilidade do solo — já foi desenvolvido em um trabalho anterior. Assim, o presente estudo concentra-se exclusivamente no desenvolvimento da solução de comunicação, armazenamento, supervisão e monitoramento remoto dos dados coletados, por meio de uma interface web integrada ao sistema.

1.4 Estrutura do trabalho

Este trabalho está organizado da seguinte forma:

O Capítulo 2 apresenta os fundamentos teóricos que embasam a pesquisa. O Capítulo 3 descreve a metodologia adotada para o desenvolvimento do sistema. O Capítulo 4 detalha a implementação do sistema supervísório, incluindo a interface web e a integração com os dispositivos de *hardware*. O Capítulo 5 analisa os resultados obtidos, destacando as contribuições e limitações do sistema proposto. O Capítulo 6 traz as considerações finais, sugerindo possíveis aprimoramentos e aplicações futuras.

2 REFERENCIAL TEÓRICO

Para a implantação de um Sistema supervisório faz-se necessário, por vezes, a integração de diversas ferramentas e serviços como sensores e atuadores, controladores, serviço de redes de comunicação, controladores e interfaces homem-máquina. Nas seguintes seções serão apresentados os conceitos envolvidos na execução deste trabalho.

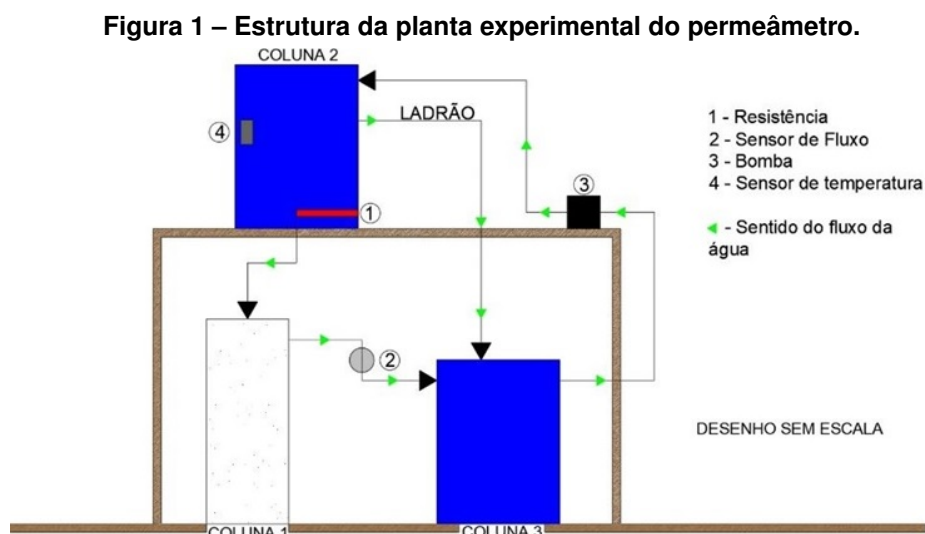
2.1 Permeâmetro e Filtros Granulares

O permeâmetro é um equipamento utilizado para medir a permeabilidade do solo, isto é, a capacidade que o solo possui de permitir a passagem de fluidos através de seus poros. Essa característica é fundamental para diversas aplicações na engenharia civil e ambiental, especialmente em obras que envolvem solos granulares, como sistemas de drenagem, fundações e estruturas de contenção.

Neste trabalho, foi utilizado um permeâmetro do tipo carga constante, conforme recomendado pela norma ABNT NBR 13292 (2021) para ensaios em solos de granulometria grossa. O sistema experimental é composto por três colunas principais:

- A **Coluna 1**, onde são colocados os solos granulares que simulam o filtro;
- A **Coluna 2**, que funciona como reservatório superior, equipado com um sistema de ladrão que mantém constante o nível de água, garantindo a carga hidráulica constante;
- A **Coluna 3**, que atua como reservatório inferior, coletando tanto a água excedente da Coluna 2 quanto a água que atravessa o solo da Coluna 1.

A estrutura geral da planta experimental está apresentada na Figura 1.



Fonte: Adaptado de Engel (2022)

Para realizar o monitoramento contínuo do ensaio, sensores eletrônicos foram integrados à planta experimental. Esses sensores são responsáveis por medir variáveis relevantes, como o nível da água no reservatório superior, a temperatura da água percolada e a vazão de saída. As medições são realizadas por meio de um microcontrolador do tipo Arduino, ao qual os sensores estão fisicamente conectados via entradas digitais.

Essa instrumentação foi inicialmente desenvolvida por Engel (2022), que propôs uma automação local baseada no microcontrolador ATmega 2560. No presente trabalho, essa base foi aprimorada com a adição de um microcontrolador ESP32 e a integração do sistema a uma plataforma de supervisão em nuvem, como será detalhado nos capítulos seguintes.

2.2 Sistemas Supervisórios

Nas últimas décadas do século XX, os sistemas supervisórios começaram a ser inseridos nos processos industriais (PAIOLA, 2015). Com a evolução da automação, a necessidade de *software* que supervisionassem os processos aumentou. No final da década de 90 já existiam diversos fabricantes de *software* de supervisão que trouxeram ferramentas padronizadas para construção de interfaces entre o operador e o processo (KATSURAGAWA, 2012; WOLF, 2015).

De acordo com Paiola (2015), um sistema supervisório é composto por telas, base de dados e drivers de comunicação. Tais telas são animadas por informações das bases de dados provenientes de diversas fontes de campo (CLPs e outros dispositivos). Apesar dos sistemas supervisórios serem utilizados amplamente pelos operadores de chão de fábrica, na lista de usuários dos sistemas existem perfis de supervisão, gerentes e até diretores das empresas. A disponibilidade de informações em tempo real e confiáveis atingiu um posto muito mais importante no universo da automação industrial, permitindo real integração com outros ambientes, sejam eles típicos do chão-de-fábrica ou mesmo gerenciais, fornecendo dados precisos para análises de processo (PAIOLA, 2015).

Sistemas SCADA têm como características, no que diz respeito à aplicação, serem utilizados em plantas onde as variáveis do processo têm que ser medidas e controladas remotamente. São sistemas que demandam uma solução para cada caso e devem ser estudados e desenvolvidos conforme a aplicação específica (JUNIOR, 2019). Um sistema SCADA obrigatoriamente possui uma IHM (interface homem-máquina) e é formado pelos três principais elementos: estação remota, protocolo de comunicação e central de controle e operações (JUNIOR, 2019), mostrados na Figura 2.

Os elementos das estações remotas são sensores, atuadores, transdutores e todos os equipamentos responsáveis por realizar as aferições e medições. A comunicação abrange tanto os meios pelos quais as informações se propagam quanto os programas e protocolos responsáveis por sua interpretação nas Centrais de Controle, por meio de elementos gráficos.

Figura 2 – Elementos de um sistema SCADA

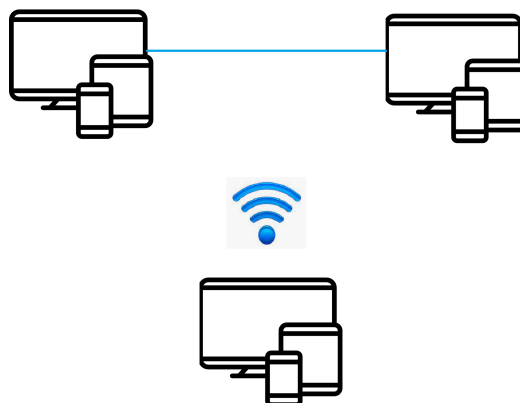


Fonte: Adaptado de JUNIOR (2019)

2.3 Comunicação

De acordo com IDEALI (2021), a conectividade em eletrônica ou computação é a capacidade de equipamentos eletrônicos comunicarem-se de forma adequada para circulação de dados e informações. Atualmente, a comunicação pode ser estabelecidas de duas maneiras: por cabos, fios de ligação, de uma ponta a outra, ou sem fio, estabelecida pela emissão de ondas eletromagnéticas, de forma que a captura desse sinal é feita por outra ponta que esteja devidamente sintonizada com essa emissão de frequência (IDEALI, 2021), a Figura 3 ilustra os dois princípios de comunicação.

Figura 3 – Princípios de comunicação



Fonte: Autoria própria (2023)

A conexão de microcontroladores com dispositivos, como sensores, microcontroladores, computadores ou até mesmo a internet, é comumente utilizada no universo da conectividade. Desta forma, diferentes tipos de protocolos se fazem necessários sempre que há dois ou mais dispositivos se comunicando (IDEALI, 2021).

Protocolos de comunicação são conjuntos de regras que formatam e processam os dados. Os dispositivos em uma rede podem usar *software* e *hardwares* muito diferentes, porém, os protocolos permitem que se comuniquem entre si independentemente dessas diferenças (FLARE, 2023).

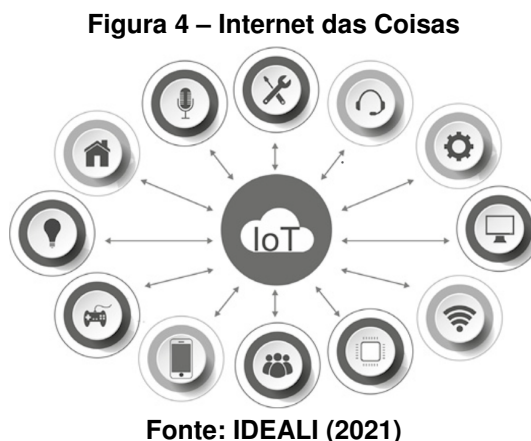
Atualmente, observa-se um forte processo de interconexão entre pequenos dispositivos, impulsionado pelo conceito de IoT (Internet das Coisas) . A IoT refere-se à integração de sensores, atuadores e outros dispositivos à internet, permitindo a coleta, transmissão e análise de dados em tempo real (IDEALI, 2021).

Nesse contexto, torna-se essencial o uso de protocolos de comunicação, que definem as regras e os formatos para a troca de informações entre os dispositivos conectados e o mundo externo.

Neste projeto, serão utilizados os protocolos I²C (para comunicação entre componentes locais) e TCP/IP (para comunicação em rede), além da tecnologia de conexão sem fio Wi-Fi, que permitirá o envio dos dados coletados para o ambiente web. As vantagens e características dessas tecnologias serão detalhadas nas próximas seções.

2.3.1 Internet da Coisas

Internet das coisas (IoT) é um conceito inovador que está crescendo rapidamente e ganhando espaço no cenário atual, onde as redes Wi-Fi e a internet são fundamentais (CORDEIRO, 2019). A internet é uma rede global que permite a comunicação e o compartilhamento de informações entre pessoas e dispositivos eletrônicos em todo o mundo. Ela conecta bilhões de dispositivos, incluindo computadores, smartphones, tablets e outros aparelhos, permitindo a troca de dados e informações em tempo real (COMER, 2016). O Wi-Fi, por sua vez, é uma tecnologia de rede sem fio que permite que dispositivos se conectem a uma rede física sem a necessidade de cabos ou fios. O Wi-Fi usa ondas de rádio para transmitir dados entre dispositivos e roteadores que estão conectados à internet (ALECRIM, 2008), a Figura 4 ilustra a interconexão de dispositivos.



A ideia central do conceito IoT é a interconexão de dispositivos (sensores, celulares, atuadores etc.), que são capazes de interagir entre si e facilitar diversas atividades do cotidiano humano (CORDEIRO, 2019; IDEALI, 2021). Nesse sentido, a conectividade é uma meta importante a ser alcançada pela IoT, e um dos protocolos de comunicação mais utilizados é o conhecido TCP/IP, detalhado na seguinte seção. (ALEXANDRE; CORTE, 2020).

2.3.2 Protocolo TCP/IP

Já no início da era da comunicação, com a variedade de sistemas operacionais e a diversidade de equipamentos para serem interconectados, houve uma necessidade de padronização dos protocolos de comunicação. A ISO (*International Organization for Standardization*) propôs o modelo OSI (*Open System Interconnection*) com sete camadas: aplicação, apresentação, sessão, transporte, rede, enlace e física (PINTO, 2010), a Figura 5 descreve a funcionalidade de cada camada .

Figura 5 – Modelo OSI

CAMADA	FUNÇÃO
7 - Aplicação	Funções Especialistas (transferência de arquivos, envio de e-mail, terminal virtual)
6 - Apresentação	Formatação dos dados, conversão de códigos e caracteres (compreensão e criptografia)
5 - Sessão	Negociação e conexão com outros nós, analogia
4 - Transporte	Oferece métodos para a entrega de dados ponto a ponto
3 - Rede	Roteamento de pacotes em uma ou várias redes
2 - Enlace	Deteção de erros
1 - Física	Transmissão e recepção dos bits brutos através do meio físico de transmissão

Fonte: Adaptado de Pinto (2010)

O TCP/IP compreende um conjunto de protocolos que regularizam a comunicação entre dispositivos em uma rede, TCP (Protocolo de Controle de Transmissão), e IP (Protocolo de Internet). O TCP/IP funciona de modo hierárquico, sendo os protocolos de nível superior suportados por protocolos de nível inferior.

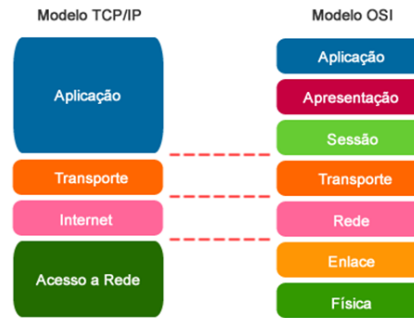
Um protocolo de comunicação não precisa, necessariamente, utilizar as sete camadas propostas pelo modelo OSI. A exemplo disso, o protocolo TCP/IP pode ser abstraído em quatro camadas: : camada de aplicação, camada de transporte, camada de internet e camadas física e de enlace. A Figura 6 mostra a equivalência entre cada camada do modelo OSI e o modelo TCP/IP (MARIANO; SOARES; NETO, 2020).

A seguir, serão descritas as diversas camadas utilizadas pelo protocolo TCP/IP.

2.3.2.1 Camada de aplicação

É a camada de nível mais alto, ou seja, composta por protocolos que fornecem serviços de rede diretamente aos usuários finais. Talvez o mais conhecido seja o protocolo HTTP

Figura 6 – Pilha de protocolos e suas equivalências



Fonte: dataRain (2023)

(Protocolo de Transferência de Hipertexto), usado pelos navegadores de internet (MARIANO; SOARES; NETO, 2020; IDEALI, 2021). A seguir, são apresentados exemplos de outros protocolos que operam na camada de aplicação, sendo o HTTP possivelmente o mais conhecido entre eles:

- **Hyper text transfer protocol (HTTP):** O protocolo de transferência de hipertexto é utilizado para comunicação entre servidores web e navegadores de internet, como Chrome e Firefox (IDEALI, 2021).
- **Hyper text transfer protocol secure (HTTPS) :** O protocolo de transferência de hipertexto seguro funciona de modo similar ao HTTP, com a diferença de criptografar os dados transferidos. Na verdade, trata-se de uma combinação do protocolo HTTP com o protocolo SSL (*Secure Sockets Layers*), cujo princípio básico reside no fato de que a conexão entre usuário e servidor esteja protegida (MARIANO; SOARES; NETO, 2020; PINTO, 2010; IDEALI, 2021).
- **Simple Mail Transfer Protocol (SMTP):** Protocolo simples usado para transferência de e-mails (IDEALI, 2021).
- **Dynamic Host Configuration Protocol (DHCP):** O protocolo de configuração de host dinâmico é utilizado para atribuir endereços de IP em uma rede, como em uma rede Wi-Fi doméstica. Quando um usuário se conecta à rede, ele necessita de um endereço de IP. Endereços de IP podem ser configurados manualmente ou atribuídos automaticamente por um servidor DHCP. Em geral, em uma rede Wi-Fi doméstica, o modem/roteador tem a tarefa de atribuir endereços de IP a dispositivos conectados (MARIANO; SOARES; NETO, 2020; PINTO, 2010; IDEALI, 2021).
- **Domain Name System (DNS):** O sistema de nomes de domínios traduz endereços de IP em nomes de domínio, e vice-versa (IDEALI, 2021)
- **File Transfer Protocol (FTP) :** O protocolo de transferência de arquivos é utilizado para transferir arquivos entre computadores distintos por uma rede, como a internet.

O FTP permite que um usuário acesse remotamente um servidor FTP, podendo enviar e receber arquivos de maneira fácil e segura. O protocolo é bastante utilizado por empresas e usuários individuais para compartilhar arquivos grandes ou fazer backup de dados (IDEALI, 2021).

2.3.2.2 Camada de transporte

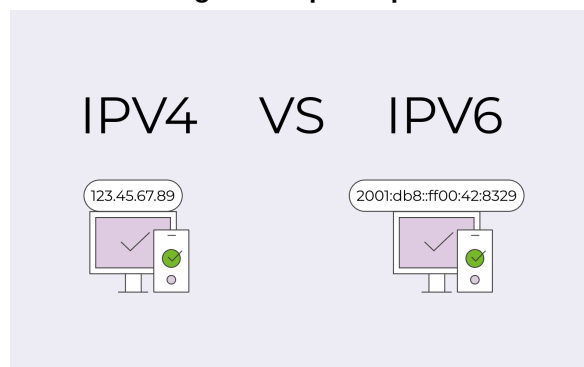
A camada de transporte é responsável por receber e transportar os dados gerados na camada de aplicação, dividindo-os em pacotes para a transmissão. Nesta camada os dois principais protocolos utilizados são o TCP e o UDP (*User Datagram Protocol*). O protocolo TCP organiza e transmite os pacotes de dados, garantindo que sejam entregues integralmente ao final da transmissão, já o protocolo UDP apenas realiza a transmissão dos dados, sem verificar se foram entregues integralmente ou retransmitir pacotes perdidos (IDEALI, 2021).

2.3.2.3 Camada de internet

A camada de internet, de acordo com IDEALI (2021) desempenha um papel crucial na comunicação em rede, sendo responsável pelo endereçamento universal dos dispositivos conectados. O endereço virtual atribuído a um dispositivo é definido pelo IP, um dos protocolos mais importantes da camada (IDEALI, 2021). Além disso, a camada de internet também utiliza outros protocolos, como o ARP (*Address Resolution Protocol*), que mapeia os endereços de rede ao *hardware*, e o RARP (*Reverse Address Resolution Protocol*), que associa um endereço MAC a um IP.

Atualmente, existem duas versões ativas do protocolo IP: o IPv4 e o IPv6. O IPv4 permite endereços de 32 bits, representados por quatro números que variam de 0 a 255 e separados por pontos. No entanto, devido à limitação de 2^{32} endereços de IP (equivalente a cerca de 4 bilhões de endereços), surgiu o IPv6, que oferece suporte a endereços de 128 bits (IDEALI, 2021). A Figura 7 ilustra os diferentes modelos de IPs existentes.

Figura 7 – Ipv4 e Ipv6



Fonte: One.com (2023)

2.3.2.4 Camada física e enlace

No TCP/IP, as camadas física e enlace se equivalem às duas primeiras camadas do modelo OSI. Nelas, são atribuídos os chamados endereços físicos (endereço MAC), que corresponde a um identificador atribuído a uma interface de rede por um fabricante. Enquanto o endereço de IP é dinâmico, isto é, pode ser alterado via *software*, não se pode alterar o endereço MAC, composto por 12 algarismos hexadecimais separados por dois-pontos (IDEALI, 2021).

2.3.3 Arquitetura IoT

Os protocolos de IoT, podem estar relacionados às suas áreas de atuação e desenvolvimento, como identificação, infraestrutura, descoberta, dados, comunicação e transporte. Um dos modelos de camadas para tratar as comunicações IoT pode ser desenvolvido pelos conceitos de camadas da internet TCP/IP, classificando-os de acordo com suas funcionalidades. Esse processo de camadas se torna um apoio para solucionar problemas no meio IoT (MASCHIETTO; VIEIRA; TORRES, 2021). A Figura 8 apresenta uma comparação entre uma camada de exemplo para IoT e a camada da internet.

Figura 8 – Comparação entre camada IoT e TCP/IP

Modelo TCP/IP	Protocolos IoT
Aplicação	HTTPS, XMPP, CoAP, MQTT, AMQP
Transporte	UDP, TCP
Internet	IPV6, 6LoWPAN, RPL
Rede / Física	Ieee802.15.4, Wifi (802.11 a/b/g/n), Ethernet (802.3), GSM, CDMA, LTE

Fonte: MASCHIETTO, VIEIRA e TORRES (2021)

Quando um dispositivo IoT é conectado via Wi-Fi, o roteador é conectado a um provedor de serviço de internet. A primeira ação relacionada à rede é obter um endereço IP. Após o dispositivo possuir um IP da rede, adquirido pela configuração manual ou automática pelo protocolo DHCP, o dispositivo está pronto para acessar à internet. Em seguida, o dispositivo deseja conectar-se a um website ou servidor, e o acesso pode ser feito por meio de um nome que pode estar diretamente ligado ao serviço que se deseja acessar (IDEALI, 2021; MASCHIETTO; VIEIRA; TORRES, 2021).

Em resumo, os protocolos utilizados em aplicações de IoT são essenciais para garantir o bom funcionamento e a segurança das redes de dispositivos interconectados. Ao seguir um modelo de camadas TCP/IP, é possível classificar os protocolos de acordo com suas funciona-

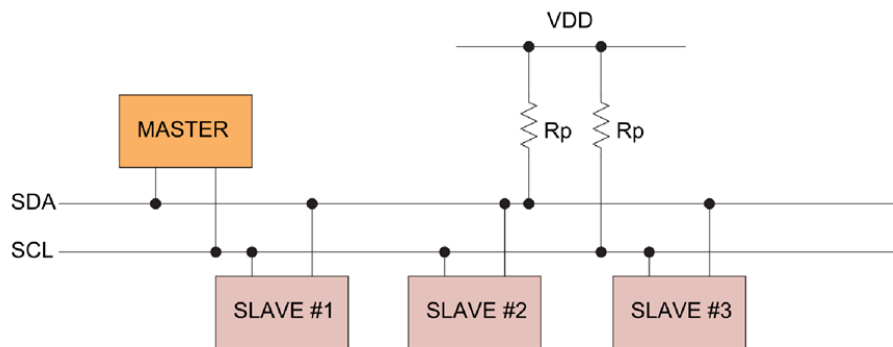
lidades, facilitando a solução de problemas no ambiente IoT. É importante compreender todos esses aspectos para garantir que as conexões IoT sejam efetivas e seguras.

2.3.4 Protocolo I²C

O protocolo de comunicação serial denominado I²C (*Inter-Integrated Circuit*), foi criado na década de 90 pela *Philips Semiconductor Division*, tendo como objetivo a conexão de microprocessadores a dispositivos controladores utilizados em equipamentos de áudio e principalmente televisores. Os estudos levaram então à criação de um barramento de 2 fios, que faz a interconexão de circuitos integrados, constituindo uma interface simples, prática e padronizada (SILVA; RICARDO; KASCHNY, 2012).

O protocolo I²C é baseado no modelo de comunicação mestre-escravo. Neste modelo, o dispositivo mestre pode inicializar a comunicação e os demais dispositivos escravos respondem enviando os dados solicitados pelo mestre ou realizar alguma ação solicitada. O barramento I²C é conectado a um processador através de apenas 2 fios. O fio denominado SCL (*Serial Clock Line*) é responsável pelo *clock* do barramento, e o fio SDA (*Serial Data Line*) pela transmissão de dados (Figura 9). Este protocolo possibilita um fluxo de dados bidirecional (IDEALI, 2021; AFZAL, 2023).

Figura 9 – Barramento I²C

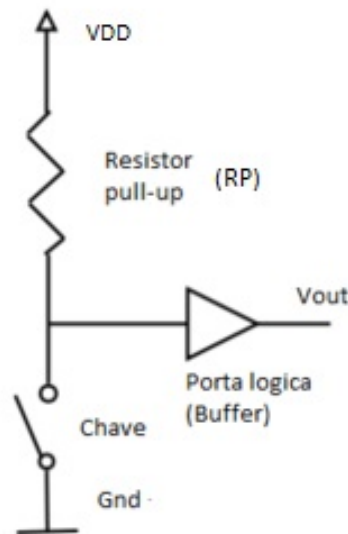


Fonte: Afzal (2023)

O barramento possui tipicamente uma alimentação VDD de 3.3V ou 5V, com dois Resistores *pull-ups* (R_p), e chaves comutadas pelos dispositivos, Figuras 9 e 10.

O microcontrolador irá ler em *V_{out}* um nível alto quando a chave não está fechada. Em outras palavras, uma pequena quantidade de corrente flui entre VDD e o pino de entrada em *V_{out}*, do modo que em *V_{out}* possua um valor próximo a VDD. Quando a chave é fechada, ele conecta o pino de entrada diretamente ao GND. A corrente flui através do resistor até o GND, de modo que *V_{out}* apresente nível baixo de tensão. A regra geral é usar um resistor de *pull-up* (R_p) que seja uma ordem de magnitude (1/10) menor do que a impedância de entrada do microcontrolador (REIS, 2016).

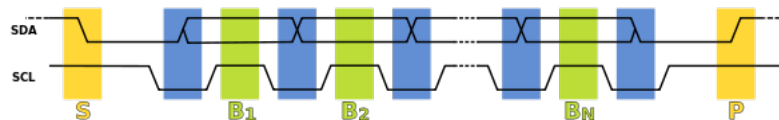
Figura 10 – Resistores Pull-Ups



Fonte: Adaptado de Oliveira (2018)

Durante o estado que a chave está aberta no barramento I²C são mantidos o valor digital alto em ambas as linhas de comunicação. Para iniciar-se a comunicação, SDA é trazido para o valor digital baixo pelo mestre. Para escrever dados no barramento, a chave da linha SCL pulsa, e a cada pulso, o valor em SDA é lido como um bit. Após SDA mudar para o nível baixo, o mestre escreve o endereço do dispositivo que ele deseja se comunicar (CAMARA, 2023).

Figura 11 – Comunicação I2C



Fonte: Adaptado de Souza (2023)

A comunicação sempre se inicia com um comando INICIAR e um comando PARAR no final, a Figura 11 ilustra o diagrama do tempo. Dessa maneira, é possível estabelecer uma comunicação entre diferentes tipos de microcontroladores.

A comunicação entre microcontroladores é crucial para o processamento de informações nas centrais de controle, como veremos na seção seguinte.

2.4 Central de Controle

A principal contribuição deste trabalho para o sistema estudado é a implementação de supervisórios web, que são ferramentas visuais que permitem o acompanhamento remoto do sistema. Para criar um supervisório, é necessário desenvolver um ambiente gráfico e criar bancos de dados. Nas seções seguintes, serão apresentados os principais conceitos necessários para a criação do ambiente gráfico e como os dados serão armazenados.

2.4.1 Armazenamento de dados em nuvem

Armazenamento refere-se ao processo de guardar informações ou dados de forma segura e acessível. O armazenamento pode ser físico, como em discos rígidos, unidades de estado sólido, CDs, DVDs, fitas magnéticas, entre outros dispositivos, ou pode ser virtual, como em serviços de armazenamento em nuvem.

O armazenamento em nuvem é um modelo de armazenamento virtual que permite que os dados sejam armazenados em servidores remotos, geralmente gerenciados por terceiros (AMAZON, 2023). Esse tipo de armazenamento é acessado pela internet e os dados são mantidos em servidores que estão localizados em diversos locais ao redor do mundo (AMAZON, 2023).

Os serviços de armazenamento em nuvem são oferecidos por empresas como Amazon, Google e Microsoft, que oferecem diferentes soluções para atender a diferentes necessidades de armazenamento e segurança. Entre os benefícios do armazenamento em nuvem estão a escalabilidade, a flexibilidade, a acessibilidade, a redundância e a segurança. A organização da armazenagem de dados deve ser baseada nas necessidades do sistema e das aplicações que utilizarão esses dados (AMAZON, 2023). É importante escolher uma estrutura que permita o acesso fácil e eficiente às informações armazenadas. A plataforma escolhida para o armazenamento de dados em nuvem é o *Firebase*, que será explicado no Capítulo 5.

2.4.2 Ambiente Gráfico em Sistemas SCADA

O ambiente gráfico em sistemas SCADA é um componente essencial para a interface homem-máquina (IHM), permitindo que operadores monitorem, analisem e controlem variáveis de processo de forma eficiente e intuitiva. A representação visual adequada dos dados operacionais facilita a identificação de anomalias, melhora a resposta a eventos críticos e contribui para a segurança e a produtividade dos sistemas automatizados.

Elementos gráficos típicos incluem:

- **Mímicos de processo:** representações visuais de componentes como válvulas, bombas, tanques e sensores;
- **Indicadores de status:** sinais visuais como cores, ícones e números que refletem o estado atual do sistema;
- **Gráficos de tendência:** utilizados para exibir o histórico ou comportamento em tempo real de variáveis como temperatura, vazão ou pressão;
- **Alarmes e notificações:** alertas visuais ou sonoros que informam situações fora dos parâmetros estabelecidos;

- **Elementos interativos:** botões, seletores e campos de entrada para envio de comandos e ajustes de operação.

Com o avanço das tecnologias web, tornou-se possível desenvolver ambientes gráficos acessíveis via navegador, ampliando a mobilidade e a flexibilidade de sistemas supervisórios. Ferramentas *no-code* e *low-code* têm ganhado destaque nesse contexto por permitirem a criação de interfaces personalizadas sem a necessidade de conhecimento avançado em programação.

Entre essas ferramentas, destaca-se o Bubble.io, uma plataforma *no-code* que permite desenvolver aplicações web com elementos visuais interativos, integração com banco de dados e autenticação de usuários. Essas características tornam esse tipo de ferramenta especialmente atrativa para projetos acadêmicos, protótipos ou aplicações de supervisão em pequena escala, onde a rapidez no desenvolvimento, o baixo custo e a simplicidade de manutenção são fatores determinantes.

3 MATERIAIS E MÉTODOS

A ênfase deste capítulo está em reportar o que e como será feito para alcançar o objetivo do trabalho. Este capítulo pode ser subdividido, inicialmente, em duas seções, sendo uma para os materiais e outra para os métodos.

3.1 Materiais

Esta seção descreve os materiais empregados no desenvolvimento do projeto, abrangendo tanto os componentes de *hardware* quanto os *softwares* utilizados na programação, integração e visualização dos dados.

Inicialmente, serão apresentados os microcontroladores utilizados, com suas principais características e funções no sistema. Em seguida, são detalhados os *softwares* empregados para a programação dos dispositivos, armazenamento e exibição das informações coletadas.

A compreensão dos materiais utilizados é fundamental para o entendimento da arquitetura proposta, pois cada componente desempenha um papel específico na aquisição, transmissão e disponibilização dos dados em tempo real.

3.1.1 Microcontroladores

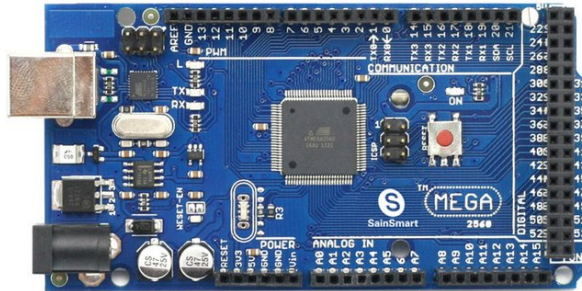
Microcontroladores são circuitos integrados que reúnem um núcleo de processador, memórias voláteis e não voláteis e diversas conexões de entrada e de saída de dados (SILVA, 2007). Um microcontrolador é capaz realizar determinadas tarefas de maneira eficaz e geralmente possui um tamanho altamente compacto (SILVA, 2007). Esse dispositivo, segundo SILVA (2007), surgiu em meados da década de 70, criado por uma equipe da Texas Instruments, derivado dos então microprocessadores criados para realizar cálculos, posteriormente tomada de decisões, foi incorporado a memórias e outras arquiteturas por um chip, evoluindo desde então e resultando no equipamento que conhecemos atualmente. Neste trabalho, serão utilizadas duas plataformas de desenvolvimento: o ESP32, equipado com o microcontrolador ESP32, e o Arduino Mega 2560, equipado com o microcontrolador ATmega2560. A seguir, abordaremos detalhadamente essas plataformas.

3.1.1.1 Arduino Mega

O Arduino Mega 2560 é uma placa de prototipagem eletrônica que utiliza o microprocessador ATmega2560 e é composta por outros componentes essenciais. Desenvolvida para facilitar a prototipagem eletrônica, essa placa é reconhecida por sua arquitetura aberta, permitindo flexibilidade nos projetos de programação, elétricos e estruturais. O Arduino Mega 2560

oferece uma ampla variedade de recursos, incluindo 54 pinos de entrada/saída digital, 16 entradas analógicas, 4 UARTs (portas seriais de *hardware*), um oscilador de cristal de 16 MHz, uma conexão USB, um conector de alimentação e um botão de reinicialização. (ARDUINO, 2023). A Figura 12 exibe o Arduino Mega 2560.

Figura 12 – Arduino Mega 2560



Fonte: FLIPFLOP (2022)

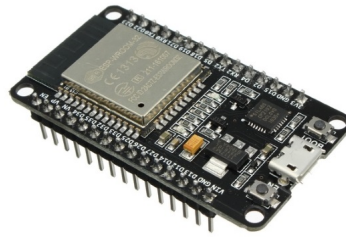
Essa ferramenta pode ser programada com o *hardware Integrated development Environment* (IDE), disponível no site do fabricante. O microcontrolador (ATmega2560) presente nesse instrumento vem pré-gravado com um *bootloader* que permite fazer *upload* de novos códigos para ele sem o uso de um programador de *hardware* externo. O Arduino Mega 2560 possui uma variedade de acessórios que ampliam sua aplicação, incluindo módulos de comunicação (como Bluetooth e Wi-Fi), módulos de sensores (como temperatura e movimento), displays (LCD e OLED), módulos de expansão de entrada/saída e motores/drivers de motor. Esses acessórios permitem conectar o Arduino a diferentes dispositivos, coletar dados do ambiente, exibir informações e controlar motores, aumentando as possibilidades de uso da placa.

3.1.1.2 ESP32

O ESP32 é um microcontrolador de baixo custo desenvolvido pela empresa chinesa Espressif Systems, tendo como fortes características sua velocidade de processamento, acessibilidade e conectividade, evidenciando essa última principalmente pela sua inteligibilidade com a conexão wi-fi (ESPRESSIF, 2023). O ESP32 é constituído por um robusto processador, foi projetado com um modelo que pode ser *single* ou *dual-core* de 32-bit (com dois núcleos físicos de processamento) que pode chegar a trabalhar com frequências de *clock* de até 240 MHz. Além disso, o ESP32 possui uma capacidade de armazenamento, significativamente maior em comparação com o popular Arduino, oferecendo o dobro de memória flash quando comparado ao modelo ATmega2560. A Figura 13 exibe o ESP32.

Existem várias opções de programação para o ESP32, como Espressif IoT Development Framework ou MicroPython, porém pela familiaridade de uso, optou-se pela IDE do Arduino. A programação do ESP32 é bastante similar à do Arduino, facilitando a transição entre as plataformas. Essas características tornam o ESP32 uma opção versátil e poderosa para projetos

Figura 13 – Placa de prototipação ESP32



Fonte: FLIPFLOP (2022)

de desenvolvimento, proporcionando flexibilidade e potencial de criação em diferentes áreas da eletrônica e da IoT.

3.1.2 *SOFTWARES*

A seção de *softwares* desempenha um papel fundamental no desenvolvimento e funcionamento deste projeto. Os *softwares* são essenciais para programar, depurar e controlar. Eles fornecem ferramentas e recursos que facilitam o processo de desenvolvimento, desde a escrita do código até a depuração e a realização de testes.

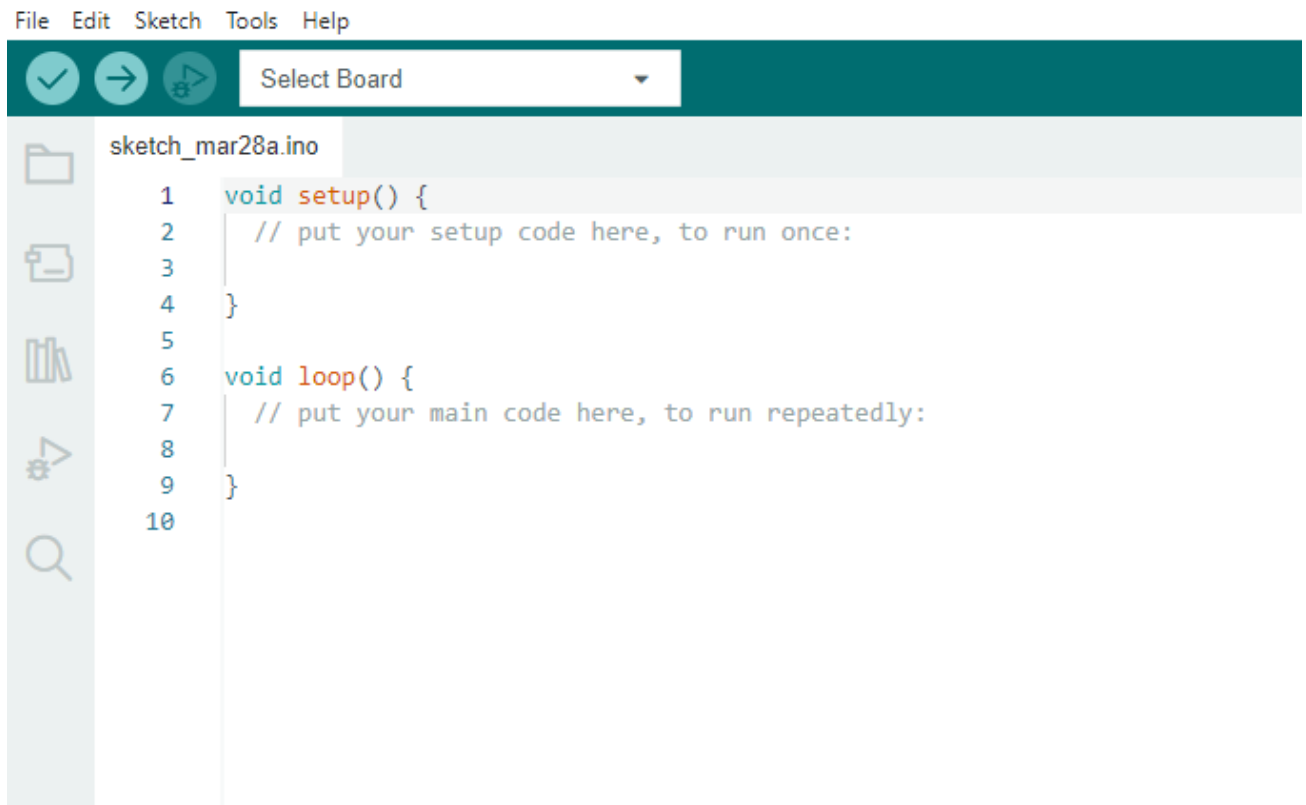
Serão destacadas as características e funcionalidades dos *softwares* que serão utilizados: a Arduino IDE, o *Firebase* e o *Bubble*. Compreender o uso dessas ferramentas é essencial para aproveitar ao máximo o potencial dessas plataformas de desenvolvimento. Vamos explorar os *softwares* utilizados, suas aplicações e como eles contribuem para o desenvolvimento.

3.1.2.1 Arduino IDE

A Arduino IDE é uma plataforma de desenvolvimento integrado que permite a programação de microcontroladores. Ela oferece recursos para escrever código, depurar e realizar testes. A IDE do Arduino oferece suporte a uma ampla variedade de placas e módulos, permitindo que desenvolvedores utilizem diferentes *hardwares* de acordo com suas necessidades. Essa flexibilidade torna a IDE do Arduino uma escolha popular para projetos de desenvolvimento. O ambiente de programação, representado na Figura 14, refere-se à interface visual fornecida pela Arduino IDE.

Para que este IDE seja utilizado para programar o ESP32 foi necessário adicionar uma Uniform Resource Locator (URL) nos Gerenciadores de Placas, permitindo o acesso à configuração de várias placas no formato “.json”, sendo possível instalar a placa “ESP32” feita pela Espressif Systems. Desta forma, o módulo ESP32 foi adicionado na lista de placas disponíveis

Figura 14 – Arduino IDE



Fonte: Arduino (2023)

e selecionado, através da porta em que estava conectado, para que os códigos pudessem ser gravados nele.

3.1.2.2 Bubble

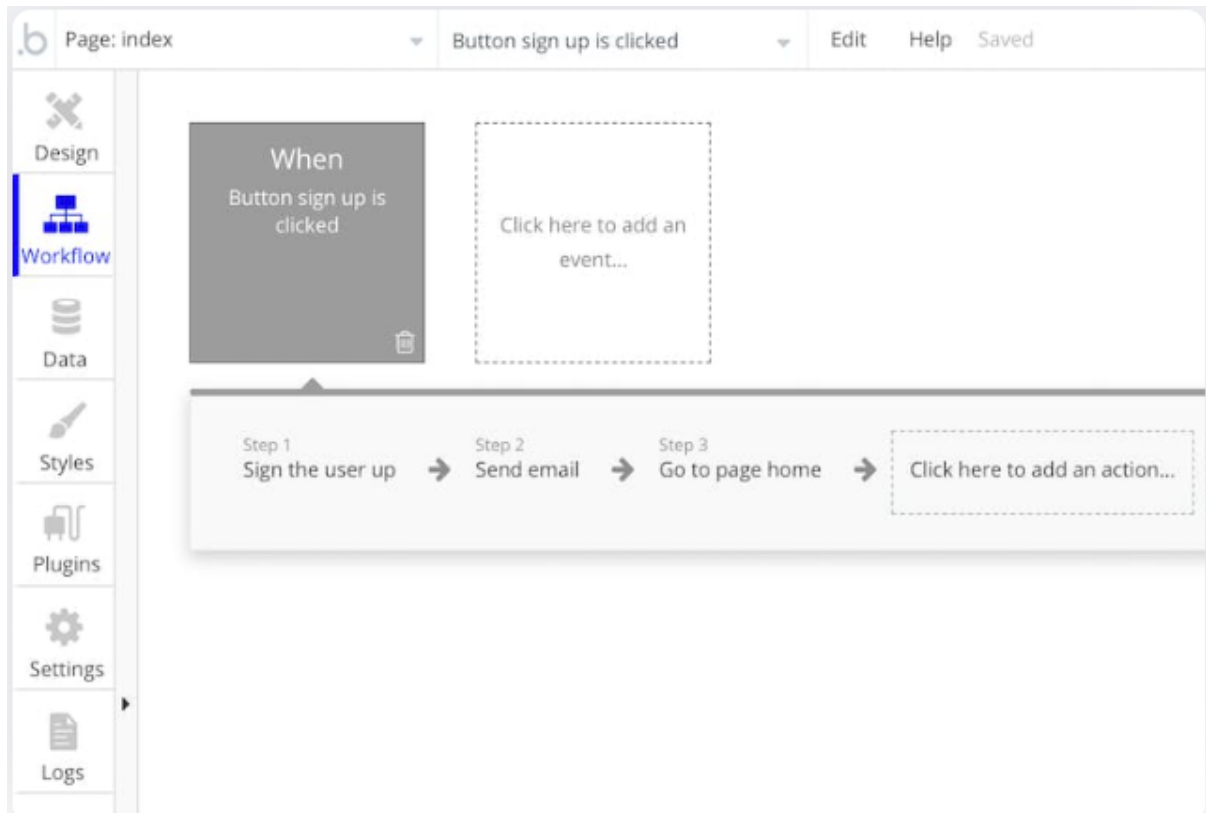
O Bubble é uma plataforma de desenvolvimento que possibilita a criação de aplicativos interativos para navegadores de desktop e dispositivos móveis, com suporte para múltiplos usuários. Essa plataforma fornece todas as ferramentas necessárias para criar um site completo, oferecendo uma abordagem visual e intuitiva para o desenvolvimento de aplicativos.

Ao utilizar o Bubble, os desenvolvedores podem criar interfaces de usuário interativas sem a necessidade de escrever código, o que simplifica o processo de desenvolvimento. A plataforma oferece recursos como arrastar e soltar elementos, criar fluxos de trabalho personalizados e conectar-se a bancos de dados externos. O ambiente de programação, representado na Figura 15, refere-se à interface visual fornecida pelo Bubble.

3.1.2.3 Firebase

O *Firebase* é uma plataforma de desenvolvimento de aplicativos web e móveis oferecida pelo Google. Ele fornece diversos serviços para ajudar os desenvolvedores a criar e aprimorar

Figura 15 – Interface do Bubble



Fonte: Bubble (2023)

seus aplicativos com eficiência. Além do *Realtime Database*, o *Firebase* oferece serviços como autenticação de usuários, armazenamento de arquivos, hospedagem de aplicativos e análise de dados.

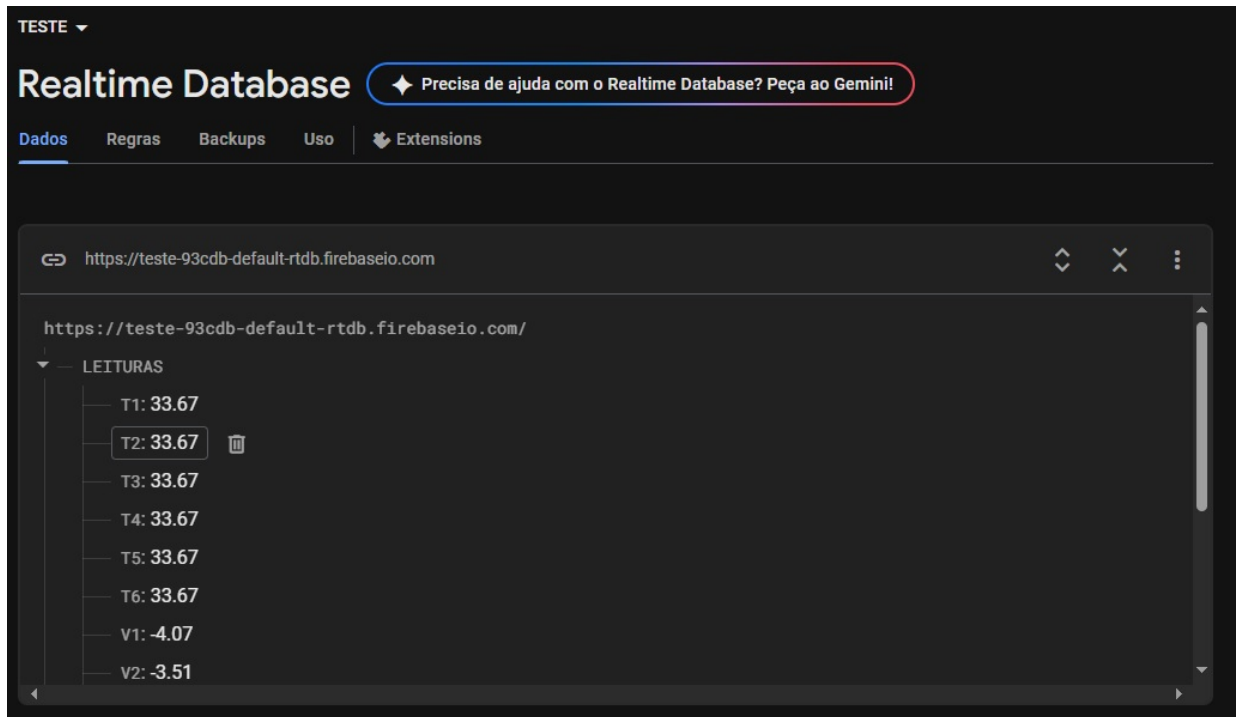
Para começar a usar o *Firebase*, é necessário criar uma conta e um novo projeto. A plataforma oferece um modelo de negócios flexível, com uma opção gratuita que possui recursos limitados, adequados para pequenos projetos e desenvolvedores individuais. Existe também uma opção paga com recursos avançados.

No *Firebase*, o fluxo de informações ocorre entre o aplicativo do cliente e os servidores do *Firebase*. Os dados são enviados do cliente para o *Firebase* por meio de uma API, onde podem ser armazenados, processados e recuperados conforme necessário.

Em relação ao *Firebase Realtime Database*, ele é uma ferramenta fundamental no projeto com o ESP32. Ele permite armazenar e recuperar dados em tempo real, facilitando a comunicação eficiente entre o dispositivo e o banco de dados hospedado na nuvem. Essa funcionalidade é especialmente útil para aplicativos de IoT, nos quais é necessário compartilhar dados entre dispositivos conectados. O *Firebase Realtime Database* também sincroniza automaticamente os dados entre os dispositivos, proporcionando uma experiência em tempo real para os usuários. A Figura 16 refere-se à interface visual fornecida pelo *Firebase*.

O *Firebase Realtime Database* desempenha um papel crucial no projeto, proporcionando uma solução para a gestão de dados em tempo real. Através dessa tecnologia, é possível

Figura 16 – Interface do *Firebase Realtime Database*



Fonte: Firebase (2023)

estabelecer uma sincronização automática dos dados entre os dispositivos, resultando em uma experiência em tempo real para os usuários.

3.2 Métodos

Este trabalho caracteriza-se como uma pesquisa aplicada, de natureza experimental e abordagem qualitativa, com foco na implementação de um sistema de comunicação entre microcontroladores e sua posterior integração com uma base de dados em nuvem, visando a visualização remota de variáveis físicas. A execução foi realizada com base em uma infraestrutura pré-existente localizada na Universidade Tecnológica Federal do Paraná (UTFPR), câmpus Toledo.

3.2.1 Arquitetura Geral do Sistema

O sistema físico da UTFPR consiste em um permeâmetro de carga constante, composto por três colunas verticais e instrumentado com sensores para monitoramento automatizado de parâmetros relevantes, como vazão, temperatura e nível de água. A coleta dessas variáveis ocorre em tempo real, permitindo a posterior transmissão dos dados para a nuvem e a visualização por meio de um sistema supervisor web.

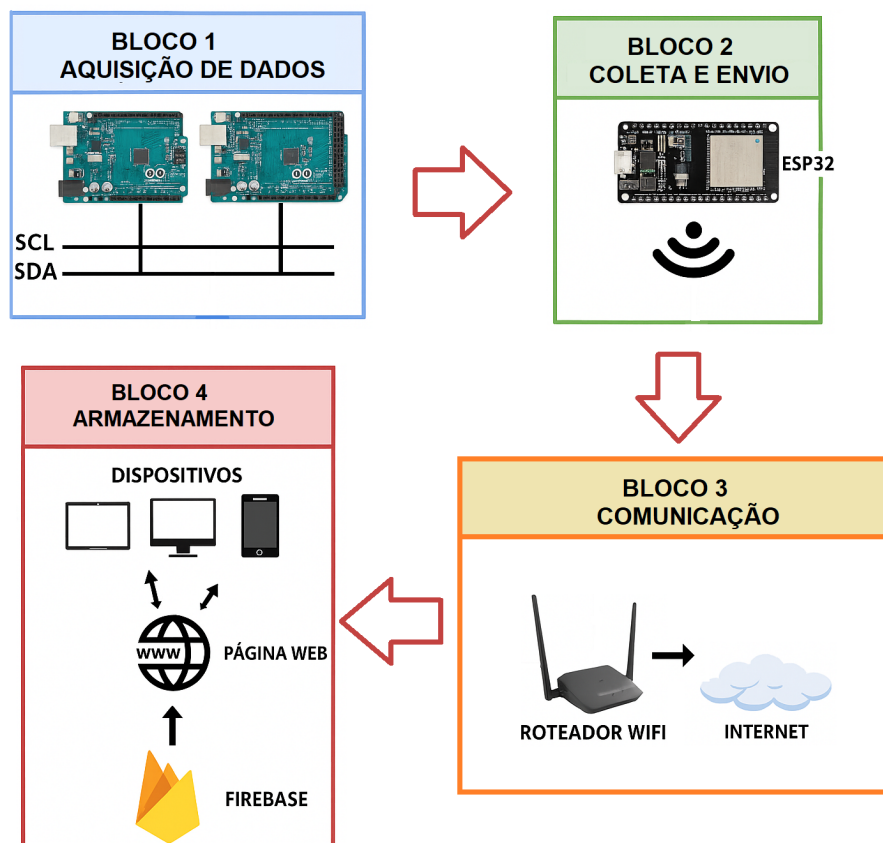
Os sensores utilizados neste projeto são:

- **Sensor de vazão YF-S201:** instalado na saída da Coluna 1 (por onde a água percola após atravessar o solo). É um sensor digital com saída em pulsos, proporcional à vazão do fluido, baseado em efeito Hall.
- **Sensor de temperatura DS18B20:** sensor digital à prova d'água, instalado na mesma tubulação de saída da Coluna 1, próximo ao sensor de vazão. Permite a medição da temperatura da água para correções no cálculo do coeficiente de permeabilidade.

Esses sensores estão conectados fisicamente ao microcontrolador Arduino Mega 2560, que realiza a leitura periódica dos dados. A comunicação entre o Arduino e o ESP32 — responsável por enviar os dados à nuvem — é feita via protocolo I²C, conforme detalhado na próxima subseção. A Figura 1 apresenta um esquema da planta experimental

Para fins deste projeto, foi adicionado um ESP32, com a finalidade exclusiva de estabelecer comunicação com o Arduino, captar as informações sensoriais e retransmiti-las à nuvem. A Figura 17 ilustra de forma esquemática essa arquitetura de comunicação entre os dispositivos.

Figura 17 – Desenho esquemático mostrando a topologia geral do projeto



Fonte: Autoria própria (2023)

- **Bloco 1 – Aquisição de Dados:** representa o conjunto de sensores conectados ao microcontrolador Arduino Mega 2560, responsável pela coleta das variáveis físicas do sistema, como vazão e temperatura. Os dados são disponibilizados via protocolo I²C.

- **Bloco 2 – Coleta e Envio:** o microcontrolador ESP32 atua como mestre na comunicação I²C, recebendo os dados do Arduino e preparando-os para transmissão via Wi-Fi.
- **Bloco 3 – Comunicação:** a transmissão dos dados ocorre por meio de uma rede Wi-Fi local, viabilizada por um roteador que estabelece a conexão entre o ESP32 e a internet.
- **Bloco 4 – Armazenamento e visualização:** os dados são enviados para o banco de dados em nuvem *Firebase Realtime Database*, que possibilita armazenamento estruturado, leitura em tempo real e integração com ferramentas externas. Por meio de uma página web criada na plataforma Bubble.io, os dados armazenados no *Firebase* podem ser acessados em tempo real por diferentes dispositivos conectados à internet, como computadores, tablets e smartphones.

Essa divisão em blocos evidencia a modularidade do sistema, facilitando sua manutenção, escalabilidade e possível adaptação para novos tipos de sensores ou plataformas de supervisão.

3.2.2 Comunicação entre Arduino e ESP32

A comunicação entre o Arduino e o ESP32 foi implementada por meio do protocolo I²C (Inter-Integrated Circuit), em conexão mestre-escravo, com o ESP32 atuando como mestre no barramento. A conexão física seguiu os padrões técnicos da interface I²C, respeitando a polaridade correta dos terminais SDA (dados) e SCL (*clock*), além da exigência de um aterramento comum entre os dispositivos.

Após a realização das conexões, foi efetuada a configuração inicial do ESP32, com a definição dos pinos e endereço do escravo I²C, bem como a inicialização da comunicação serial, conforme exemplificado no código da Figura 18

3.2.3 Transmissão de Dados à *Firebase*

Uma vez recebidos os dados via I²C, o ESP32 os transmite imediatamente, sem armazenamento local, para a plataforma *Firebase Realtime Database*, utilizando a biblioteca *IOXhop_-FirebaseESP32*. Esta biblioteca abstrai as camadas de autenticação e transporte, facilitando o envio de dados à nuvem, conforme mostra o código da Figura 19.

Embora a documentação interna da biblioteca oculte os detalhes do protocolo, supõe-se o uso de requisições HTTP ou HTTPS para a interação com a base de dados do *Firebase*, mediante o fornecimento de chave de autenticação (*API Key*) e URL do projeto *Firebase*.

O envio é realizado em formato JSON, com os campos representando os valores de vazão e temperatura, conforme ilustrado na Figura 20

Figura 18 – Código configuração comunicação I2C

```

1  #include <Wire.h>           // Inclui biblioteca para comunicação I2C
2
3  const int I2C_ADRESS = 8;   // Endereço I2C do dispositivo escravo
4
5  void setup() {
6      Wire.begin();           // Inicia o barramento I2C como mestre
7      Serial.begin(115200);   // Inicializa serial
8  }
9
10 void loop() {
11     Wire.requestFrom(I2C_ADRESS, 7*6) // Solicita 6 valores de 7 bytes cada
12     while(Wire.available()) {
13         char c = Wire.read();
14         Serial.print(c);
15     }
16 }

```

Fonte: Autoria propria (2025)

3.2.4 Visualização dos Dados

Com o objetivo de permitir a consulta remota dos dados armazenados, foi desenvolvido um painel de visualização em ambiente web por meio da plataforma Bubble.io, solução que permite a construção de *frontends* sem a necessidade de codificação tradicional.

Neste painel, o usuário tem acesso, de forma visual, aos valores armazenados de vazão e temperatura em tempo real, bem como à possibilidade de exportação dos dados históricos. A interface não é interativa no sentido de permitir alterações pelo usuário, sendo seu propósito exclusivamente informativo.

A Figura 21 apresenta a interface da página web desenvolvida para visualização das variáveis monitoradas no permeâmetro. Nela, são representadas duas plantas experimentais (Permeâmetro 1 e Permeâmetro 2), ambas compostas por três tanques conectados por tubulações.

Em cada planta, os componentes principais são:

- **Tanque superior:** representa o reservatório de carga constante (Coluna 2), onde a água é mantida em nível estável por meio de um sistema de ladrão.
- **Tanque vertical (central):** contém o solo granular e simula o filtro (Coluna 1). A água percola verticalmente através desse material.
- **Tanque inferior:** funciona como o reservatório de coleta (Coluna 3), recebendo tanto a água que atravessa o solo quanto a água excedente do sistema.

Figura 19 – Código configuração comunicação com *Firebase*

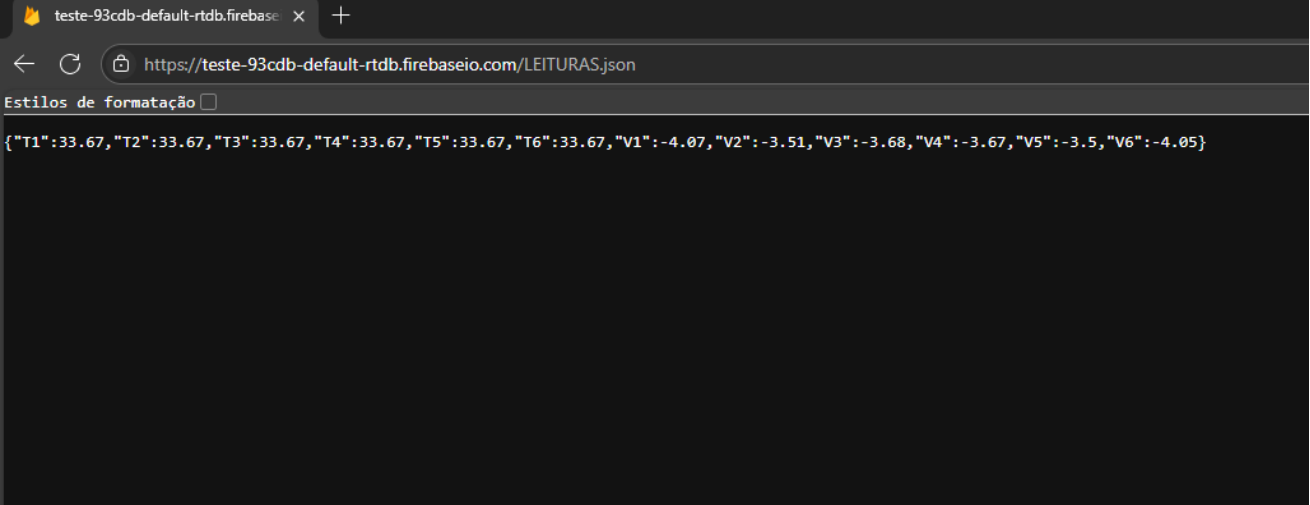
```

1  #include <WiFi.h> // Biblioteca para conectar o ESP32 ao Wi-Fi
2  #include <IOXhop_FirebaseESP32.h> // Biblioteca de autenticação do Firebase
3  #include <ArduinoJson.h> // Biblioteca para formatar JSON (usar versão 5)
4  #include <Wire.h> // Comunicação I2C
5
6  // Dados do Firebase (substitua pelo seu projeto)
7  #define FIREBASE_HOST "https://SEU_PROJETO.firebaseio.com/"
8  #define FIREBASE_AUTH "CHAVE_DE_AUTENTICACAO_DO_FIREBASE"
9  // Endereço do Arduino na comunicação I2C
10 #define ARDUINO_I2C_ADDRESS 0x08
11 void setup() {
12     Serial.begin(9600); // Inicia comunicação serial para monitoramento
13     Wire.begin(); // Inicia barramento I2C
14     // Conecta ao Wi-Fi
15     WiFi.begin("NOME_DA_SUA_REDE", "SENHA_DA_SUA_REDE");
16     while (WiFi.status() != WL_CONNECTED) {
17         delay(300);
18     }
19     // Conecta ao Firebase
20     Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
21     Serial.println("Conectado ao Firebase");
22 void loop() {
23     char buffer[10]; // Array para armazenar os valores recebidos via I2C
24     // Requisição de dados ao Arduino pelo barramento I2C
25     Wire.requestFrom(ARDUINO_I2C_ADDRESS, sizeof(buffer));
26     for (int i = 0; i < sizeof(buffer); i++) {
27         buffer[i] = Wire.read();
28     }
29     float valor = atof(buffer); // Converte string para float
30     // Exemplo de envio de dados para o Firebase
31     Firebase.setFloat("/sensor/valor", valor);
32     delay(3000); // Aguarda antes de repetir
33 }

```

Fonte: Autoria propria (2025)

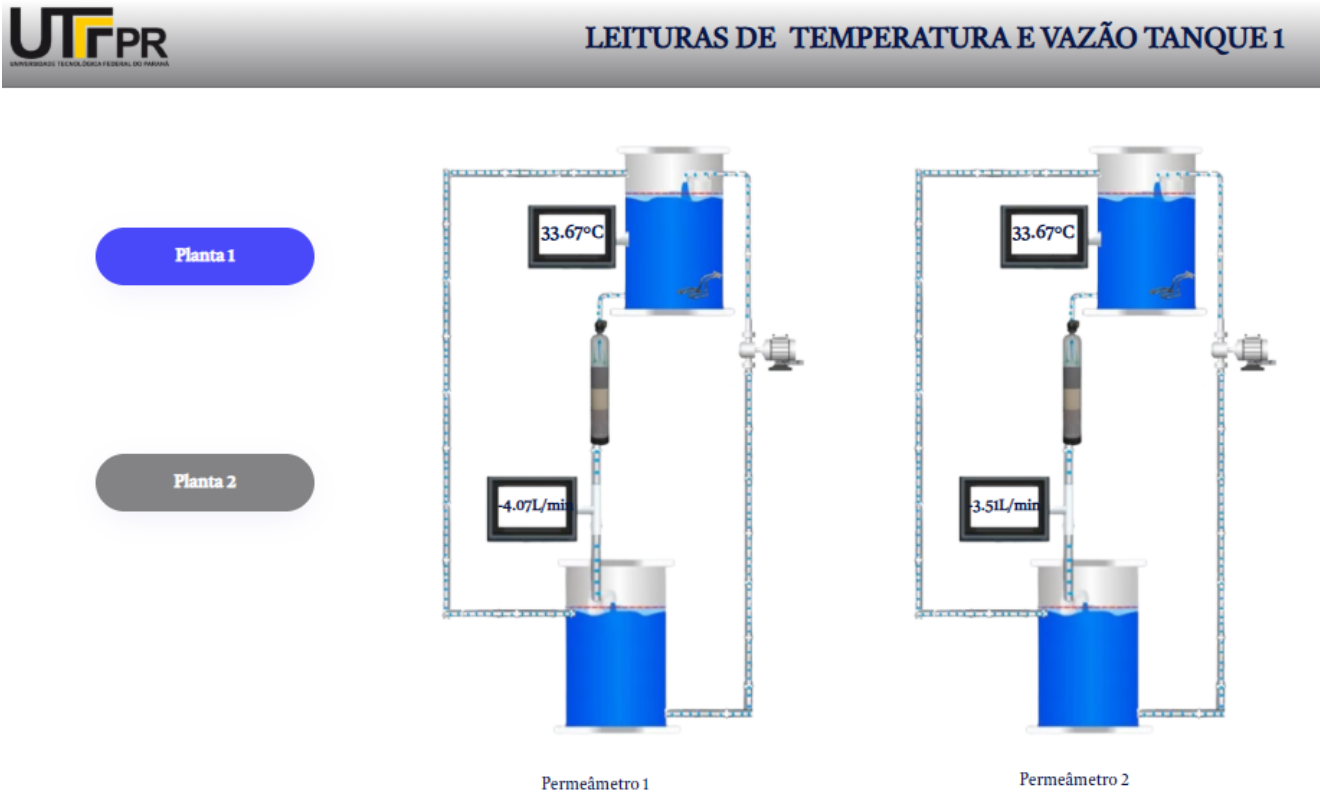
Figura 20 – Formato dos dados enviados JSON



The screenshot shows a web browser window with the address bar displaying `https://teste-93cdb-default-rtdb.firebaseio.com/LEITURAS.json`. Below the address bar, there is a section for "Estilos de formatação" (Formatting styles) with a dropdown menu. The main content area of the browser displays a JSON array of objects, each representing a sensor reading with a timestamp and a value. The JSON is: `{"T1":33.67,"T2":33.67,"T3":33.67,"T4":33.67,"T5":33.67,"T6":33.67,"V1":-4.07,"V2":-3.51,"V3":-3.68,"V4":-3.67,"V5":-3.5,"V6":-4.05}`.

Fonte: Autoria propria (2025)

Figura 21 – Tela de monitoramento dos dados



Fonte: Autoria própria (2025)

- **Sensores digitais:** instalados nas saídas da Coluna 1. Cada permeâmetro conta com um sensor de temperatura (acima) e um sensor de vazão (abaixo), cujas leituras são apresentadas em tempo real.
- **Indicadores de dados:** mostram a temperatura da água (em °C) e a vazão instantânea (em L/min), conforme medido pelos sensores.
- **Botões “Planta 1” e “Planta 2”:** possibilitam alternar entre os dados de diferentes unidades experimentais. A estrutura visual permanece a mesma, sendo apenas os dados exibidos que são atualizados conforme a planta selecionada.

A interface foi construída na plataforma Bubble.io e recebe os dados por meio do banco de dados *Firebase*, permitindo acesso remoto e visualização simplificada por dispositivos conectados à internet.

3.2.5 Considerações Técnicas

O desenvolvimento da lógica de controle e comunicação do ESP32 foi realizado na IDE Arduino, com as bibliotecas específicas para I²C e *Firebase* devidamente importadas e configu-

radas. O upload do código foi feito via cabo USB, e os testes iniciais foram conduzidos com o uso do monitor serial da IDE, para depuração em tempo real.

4 RESULTADOS E DISCUSSÕES

A implementação do sistema proposto demonstrou-se funcional e coerente com os objetivos definidos no escopo deste trabalho. Os testes realizados em ambiente controlado indicaram que a arquitetura empregada — composta pelo Arduino Mega 2560, ESP32, *Firebase* e interface web desenvolvida na plataforma Bubble.io — cumpriu as funções de aquisição, transmissão, armazenamento e visualização dos dados em tempo real.

Durante o processo de validação, observou-se que o protocolo I²C proporcionou comunicação estável entre o Arduino e o ESP32, com latência compatível com o intervalo de aquisição programado definido em 5 minutos. A transmissão dos dados ocorreu com sucesso em todos os ciclos de leitura testados.

A biblioteca *IOXhop_FirebaseESP32* mostrou-se eficiente na abstração das rotinas de autenticação e envio de dados ao *Firebase*. O envio em formato JSON garantiu clareza e consistência na estrutura dos dados, facilitando sua manipulação tanto na visualização quanto na exportação.

4.1 Visualização e Interação com os Dados

A visualização dos dados foi realizada por meio de uma interface desenvolvida na plataforma Bubble.io. Essa interface foi projetada com foco em simplicidade e objetividade, permitindo ao usuário acessar, em tempo real, as informações de vazão e temperatura coletadas pelos sensores. A conexão com o *Firebase* é feita diretamente pela plataforma, que utiliza chamadas HTTP para acessar os dados armazenados no *Realtime Database*.

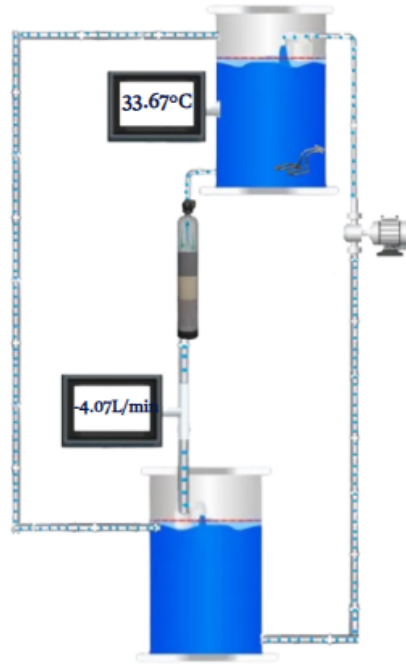
A Figura 21 mostra a tela principal da interface, e A Figura 22 mostra o permeâmetro individual, com a exibição dos valores atualizados em tempo real. A atualização ocorre automaticamente à medida que novos dados são enviados ao *Firebase*, sem necessidade de recarregamento da página.

Na parte superior da tela, é possível visualizar os valores atuais de vazão (em l/min) e temperatura (em celsius), apresentados em caixas destacadas. Logo abaixo, há um painel com a visualização de dados históricos, onde os registros são exibidos em ordem cronológica.

Para a configuração inicial do ambiente, foi necessário integrar o *Firebase* à plataforma Bubble utilizando o plugin de conexão com APIs externas. Esse plugin permite mapear os caminhos de acesso dentro do *Realtime Database*, realizar testes de autenticação e estruturar os elementos visuais conforme os campos de dados.

Embora a interface atual não permita interações com o sistema físico, como o envio de comandos ou ajuste de parâmetros, ela cumpre seu papel como ferramenta de monitoramento remoto, com possibilidade de expansão futura para aplicações mais complexas.

Figura 22 – Circuito individual tela de monitoramento



Fonte: Autoria própria (2025)

4.2 Exportação de Dados e Armazenamento Local

A exportação automatizada dos dados para o *Google Sheets* foi implementada por meio de um *script* escrito em *Google Apps Script*. Esse *script* realiza leituras periódicas do banco de dados *Firebase* e grava os valores em planilhas do *Google Drive*. A Figura 23 mostra um exemplo dos dados registrados.

Figura 23 – Exemplo de dados armazenados no Google Sheets

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	T1	T2	T3	T4	T5	T6	V1	V2	V3	V4	V5	V6	timestamp
2	1	2	3	4	5	6	1	2	3	4	5	6	02/02/2025 11:36:52
3	1	2	3	4	5	6	1	2	3	4	5	6	02/02/2025 11:38:18
4	1	2	3	4	5	6	1	2	3	4	5	6	02/02/2025 11:43:17
5	1	2	3	4	5	6	1	2	3	4	5	6	02/02/2025 11:48:18
6	1	2	3	4	5	6	1	2	3	4	5	6	02/02/2025 11:53:18
7	1	2	3	4	5	6	1	2	3	4	5	6	02/02/2025 11:58:18
8	1	2	3	4	5	6	1	2	3	4	5	6	02/02/2025 12:03:19
9	1	2	3	4	5	6	1	2	3	4	5	6	02/02/2025 12:08:18
10	1	2	3	4	5	6	1	2	3	4	5	6	02/02/2025 12:13:18
11	1	2	3	4	5	6	1	2	3	4	5	6	02/02/2025 12:18:18
12	137,72	50,98	28,04	138,04	138,04	138,04	128,36	47,52	26,13	126,13	138,04	138,04	02/02/2025 12:23:19
13	333,24	123,35	67,84	337,84	337,84	337,84	333,22	123,34	67,83	337,84	337,84	337,84	02/02/2025 12:28:18
14	337,74	337,74	337,74	337,74	337,74	337,74	337,74	337,74	337,74	337,74	337,74	337,74	02/02/2025 12:33:18
15	12,5	13,45	16,89	36,89	36,89	36,89	12,5	13,45	16,89	36,89	36,89	36,89	02/02/2025 12:38:18
16	12,5	13,45	16,89	36,89	36,89	36,89	12,5	13,45	16,89	36,89	36,89	36,89	02/02/2025 12:43:18
17	12,5	13,45	16,89	36,89	36,89	36,89	12,5	13,45	16,89	36,89	36,89	36,89	02/02/2025 12:48:19
18	12,5	13,45	16,89	36,89	36,89	36,89	12,5	13,45	16,89	36,89	36,89	36,89	02/02/2025 12:53:19
19	12,5	13,45	16,89	36,89	36,89	36,89	12,5	13,45	16,89	36,89	36,89	36,89	02/02/2025 12:58:18
20	12,5	13,45	16,89	36,89	36,89	36,89	12,5	13,45	16,89	36,89	36,89	36,89	02/02/2025 13:03:18
21	12,5	13,45	16,89	36,89	36,89	36,89	12,5	13,45	16,89	36,89	36,89	36,89	02/02/2025 13:08:18

Fonte: Autoria própria (2025)

Essa abordagem permite o armazenamento local para fins de auditoria, análise estatística ou relatórios acadêmicos. A integração com o *Google Sheets* facilita o acesso e a organização dos dados, oferecendo uma alternativa prática para o acompanhamento das informações coletadas.

4.3 Limitações e Possibilidades de Expansão

Durante os testes, não foram identificadas falhas críticas de operação. No entanto, a dependência exclusiva da conexão Wi-Fi representa um ponto de fragilidade. Uma perda temporária de sinal compromete a atualização em tempo real, o que pode ser resolvido com a inclusão de um armazenamento temporário (buffer) no próprio ESP32, capaz de guardar os dados localmente até o restabelecimento da conexão.

Além disso, melhorias na interface são sugeridas, como:

- Filtros por intervalo de tempo (última hora, último dia, última semana);
- Geração automática de gráficos de tendência;
- Exportação segmentada por datas ou variáveis;
- Inclusão de alertas visuais ou sonoros para eventos críticos.

Essas melhorias podem aumentar significativamente o valor do sistema para uso acadêmico e industrial.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de um sistema supervisor baseado em placas Arduino e ESP32, integrado à plataforma *Firebase*, para a coleta, armazenamento e visualização em tempo real dos dados de um permeâmetro individual. A solução proposta demonstrou ser uma alternativa eficiente para o monitoramento automático da permeabilidade de solos granulares. A interface web desenvolvida permitiu acesso remoto simples e ágil aos dados, facilitando a análise e o acompanhamento do ensaio. Os testes realizados indicaram comunicação estável e confiável entre os dispositivos, bem como a viabilidade da exportação dos dados para ferramentas externas utilizadas em análises complementares.

Durante o desenvolvimento, alguns desafios práticos foram identificados, como a dependência da conectividade Wi-Fi para a transmissão contínua de dados e a falta de energia elétrica, que pode comprometer momentaneamente o funcionamento do sistema. Essas limitações evidenciam a necessidade de mecanismos de tolerância a falhas para garantir a integridade e continuidade do monitoramento.

Como trabalhos futuros, sugere-se a expansão da arquitetura proposta com a implementação de soluções como o uso de memória local no ESP32 para armazenamento temporário dos dados em caso de perda de conexão com a internet ou de energia, aumentando a robustez e a confiabilidade do sistema. Além disso, a substituição do *Firebase* por bancos de dados mais escaláveis, como o InfluxDB ou o MongoDB, poderá proporcionar melhor desempenho e maior flexibilidade na manipulação de grandes volumes de dados.

Recomenda-se ainda a inclusão de novos sensores e o uso de protocolos de comunicação alternativos, como o MQTT, visando reduzir a latência e aumentar a eficiência do envio de dados. Por fim, a evolução da interface web, com recursos interativos, geração de gráficos dinâmicos e filtros por período, poderá enriquecer a experiência do usuário e ampliar as possibilidades de aplicação do sistema em ambientes acadêmicos e industriais

REFERÊNCIAS

- AFZAL, S. I2c primer: What is i2c? Fevereiro 2023. Disponível em: <https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-1.html>.
- ALECRIM, E. **O que é Wi-Fi? (conceito e versões)**. [S.l.], 2008. 1 p. Disponível em: <https://www.infowester.com/wifi.php>. Acesso em: 18 de Março de 2023.
- ALEXANDRE, P.; CORTE, B. D. Automação residencial via web e app utilizando módulos wi-fi esp8266 em conjunto com sensores. p. 75, 2020.
- AMAZON. **O que é o armazenamento em nuvem?** [S.l.], 2023. 1 p. Disponível em: <https://aws.amazon.com/pt/what-is/cloud-storage>. Acesso em: 25 de Março de 2023.
- ARDUINO. **Mega 2560 Rev3**. [S.l.], 2023. 1 p. Disponível em: <https://docs.arduino.cc/hardware/mega-2560>. Acesso em: 19 de Março de 2023.
- ASIMOV, I. **Eu, Robô**. [S.l.]: Gnome Press, 1950. As Três Leis da Robótica.
- BUBBLE. **Bubble.io**. [S.l.], 2023. 1 p. Disponível em: <https://bubble.io/home/apps>. Acesso em: 09 de Maio de 2023.
- CAMARA, R. Protocolo i2c. 2023. Disponível em: <http://www.univasf.edu.br/~romulo.camara/novo/wp-content/uploads/2013/11/Barramento-e-Protocolo-I2C.pdf>.
- CARVALHO, E. F. G. **Dimensionamento, Automação, Aquisição de Dados e Controle Supervisorio de um Trocador de Calor do Tipo Casco e Tubo**. 2018. 49 p. Monografia (Graduação) — Universidade Federal do Rio Grande do Norte, Natal - RN, 2018.
- COMER, D. E. **Redes de computadores e internet**. [s.n.], 2016. 73 p. ISBN 9788582603727. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000009000&lang=pt-br&site=eds-live&scope=site>.
- CORDEIRO, M. V. de S. Automação residencial via web e app utilizando módulos wi-fi esp8266 em conjunto com sensores. p. 13, 2019.
- DATARAIN. **Qual a diferença entre o modelo OSI e o modelo TCP/IP?** [S.l.], 2023. 1 p. Disponível em: <https://www.datarain.com.br/blog/qual-diferenca-entre-modelo-osi-e-modelo-tcpip/>. Acesso em: 25 de Março de 2023.
- ENGEL, L. E. **Instrumentação Eletrônica Aplicada em Permeâmetros para Estudos de Condutividade Hidráulica em Solos Granulares**. 2022. Trabalho de Conclusão de Curso — Universidade Tecnológica Federal do Paraná (UTFPR), Toledo, 2022.
- ESPRESSIF. **ESP32**. [S.l.], 2023. 1 p. Disponível em: <https://www.espressif.com/en/products/socs/esp32>. Acesso em: 07 de Maio de 2023.
- FIREBASE. [S.l.], 2023. 1 p. Disponível em: <https://firebase.google.com/docs?hl=pt-br>. Acesso em: 09 de Maio de 2023.
- FLARE, C. O que é um protocolo de rede? Janeiro 2023. Disponível em: <https://www.cloudflare.com/pt-br/learning/network-layer/what-is-a-protocol/>.
- FLIPFLOP. **Módulo WiFi ESP32 Bluetooth**. [S.l.], 2022. 1 p. Disponível em: <https://www.filipeflop.com/produto/modulo-wifi-esp32-bluetooth/>. Acesso em: 18 de Novembro de 2022.

- IDEALI, W. **Conectividade em Automação e IoT: Protocolos I2C, SPI, USB, TCP-IP entre outros. Funcionalidade e interligação para automação e ToT.** [S.l.]: Editora Alta Books, 2021. E-book. ISBN 9786555202564, 2021.
- JUNIOR, E. **Introdução a Sistemas de Supervisão, Controle e Aquisição de Dados.** [S.l.]: Editora Alta Books, 2019. E-book. ISBN 9788550807744, 2019.
- KATSURAGAWA, T. **Automação industrial: sistemas supervisórios e controle de processos.** São Paulo: Editora Érica, 2012.
- LUGLI, A. B.; SANTOS, M. M. D. **REDES INDUSTRIAIS PARA AUTOMAÇÃO INDUSTRIAL - AS-I, PROFIBUS E PROFINET.** 2. ed. Pinheiros - SP: Editora Saraiva, 2019. E-book. ISBN 9788536532042, 2019.
- MARIANO, D. C. B.; SOARES, J. A.; NETO, R. M. **Infraestrutura de TI.** [s.n.], 2020. 63 p. ISBN 9786556900209. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786556900209/>.
- MASCHIETTO, L. G.; VIEIRA, A. L. N.; TORRES, F. E. **Arquitetura e Infraestrutura de IoT.** [s.n.], 2021. 73 p. ISBN 9786556901947. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786556901947/>.
- OLIVEIRA, R. C. de. **Conheça o Resistor de Pull Down.** [S.l.], 2018. 1 p. Disponível em: <https://embarcados.com.br/resistor-de-pull-down/>. Acesso em: 04 de Maio de 2023.
- ONE.COM. **O que é IPv6?** [S.l.], 2023. 1 p. Disponível em: <https://www.one.com/pt/alojamento/o-que-e-ipv6>. Acesso em: 25 de Março de 2023.
- PAIOLA, C. E. G. O papel do supervisório no atual contexto tecnológico. p. 7–18, 2015.
- PINTO, P. **Redes – Sabe o que é o modelo OSI?** [S.l.], 2010. 1 p. Disponível em: <https://pplware.sapo.pt/tutoriais/networking/redes-sabe-o-que-e-o-modelo-osi/>. Acesso em: 25 de Março de 2023.
- REIS, F. dos. Eletrônica – o que são resistores de pull-up. Abril 2016. Disponível em: <http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/eletronica-o-que-sao-resistores-de-pull-up/>.
- SILVA, R. A. **Programando Microcontroladores Pic - Linguagem C.** [S.l.: s.n.], 2007. 21 p. ISBN 8599823043.
- SILVA Ítallo S. L. E.; RICARDO, J.; KASCHNY, A. **Aplicações do protocolo I 2 C em sistemas microcontrolados.** [S.l.: s.n.], 2012. ISBN 9788562830105.
- SOUZA, F. **Comunicação I2C.** [S.l.], 2023. 1 p. Disponível em: <https://embarcados.com.br/comunicacao-i2c/>. Acesso em: 04 de Maio de 2023.
- WOLF, M. **Introdução a sistemas SCADA e automação industrial.** Rio de Janeiro: LTC, 2015.