

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
FACULDADE DE ENGENHARIA - CAMPUS DE ILHA SOLTEIRA
CURSO DE ENGENHARIA ELÉTRICA

GUILHERME AUGUSTO MARTINS

**AUTOMAÇÃO SUPERVISÓRIA IOT PARA TELEMETRIA DE TEMPERATURA
DE ÁGUA INDUSTRIAL UTILIZANDO ESP32**

Ilha Solteira – SP

2023

GUILHERME AUGUSTO MARTINS

**AUTOMAÇÃO SUPERVISÓRIA IOT PARA TELEMETRIA DE TEMPERATURA
DE ÁGUA INDUSTRIAL UTILIZANDO ESP32**

Trabalho de graduação apresentado à Faculdade de Engenharia do Campus de Ilha Solteira - UNESP como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Prof. Dr. Carlos Antônio Alves
Orientador

Ilha Solteira
2023

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

Martins, Guilherme Augusto.
M386a Automação supervisória IoT para telemetria de temperatura de água industrial utilizando ESP32 / Guilherme Augusto Martins. -- Ilha Solteira: [s.n.], 2023
51 f. : il.

Trabalho de conclusão de curso (Graduação em Engenharia Elétrica) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2023

Orientador: Carlos Antônio Alves

Inclui bibliografia

1. Internet das coisas. 2. Sensoriamento de temperatura. 3. Banco de dados.

ATA DE DEFESA DE TRABALHO DE GRADUAÇÃO

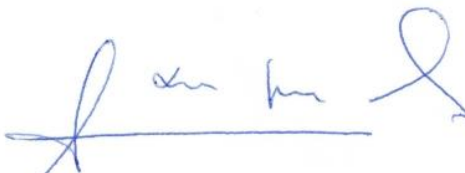
Aos dezoito dias do mês de dezembro do ano de dois mil e três, o discente **Guilherme Augusto Martins**, matriculado sob o nº 171052986, tendo como banca examinadora o seu orientador, o *Prof. Dr. Carlos Antonio Alves*, o *Prof. Dr. Alexandre Cesar Rodrigues da Silva* e o *Dr. Lucas do Carmo Yamaguti*, apresentou o Trabalho de Graduação intitulado "Automação Supervisória lot Para Telemetria De Temperatura De Água Industrial Utilizando Esp32", obtendo a nota NOVE (9.0) e conceito Aprovado.



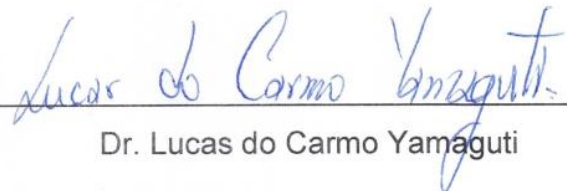
Prof. Dr. Carlos Antonio Alves
- Orientador -



Guilherme Augusto Martins
- Discente -



Prof. Dr. Alexandre Cesar Rodrigues da Silva
- Membro da Banca -



Dr. Lucas do Carmo Yamaguti
- Membro da Banca -

AGRADECIMENTOS

Dedico meus agradecimentos:

À minha família, em especial à minha mãe, Valdira, meu pai, Dijalma, e minha namorada e companheira de vida a mais de 8 anos, Ingrid, por todo apoio e incentivo durante minha caminhada acadêmica e por todos os ensinamentos.

A todos os meus amigos da turma 2017/1 do curso de Engenharia Elétrica, que compartilharam comigo momentos inesquecíveis, de muitas dificuldades e aprendizados e que levarei para sempre em minha vida.

A todos os meus amigos e estudantes da UNESP, que compartilharam de suas experiências no meu caminhar.

Aos queridos professores, técnicos de laboratório, e funcionários da UNESP, que contribuíram para minha formação. Em especial ao professor Dr^o. Carlos Antônio Alves, que aceitou me orientar na realização desse trabalho e me orientou em todo processo.

RESUMO

Todo processo que ocorre em ambiente industrial produz informações que são necessárias para análises, a fim de ter o controle do processo e monitorar com frequência o que acontece na indústria. Muitas vezes esses tipos de processos industriais são monitorados de forma física, ou seja, é necessário ter uma pessoa responsável em ir até o local, analisar e documentar os dados encontrados, o que consome tempo de trabalho do colaborador, que poderia estar desempenhando seu tempo em outra atividade, assim como isso se torna economicamente não rentável para a empresa. Este projeto tem como objetivo principal utilizar o controlador ESP32 junto ao sensor de temperatura tipo sonda para coletar as informações da água industrial em empresa que produz peças automotivas de estanqueidade, e armazenar via internet *WiFi*. Também visa abordar o armazenamento em banco de dados e posterior análise dos dados coletados via *WiFi*, para aplicação em sistema de monitoramento para o setor de manutenção industrial. Assim, possibilita-se a elaboração do circuito eletrônico do projeto em bancada e fazer a prova do conceito. E por fim, estudar a viabilidade de aplicação do projeto de baixo custo em ambiente industrial.

Palavras-chave: Internet das Coisas, Sensoriamento de temperatura, Banco de Dados.

ABSTRACT

Every process that occurs in the industrial environment produces information that is necessary to be stored and analyzed, in order to have control of the process and frequently monitor what happens in the industry. Often these types of industrial processes are physically monitored, that is, it is necessary to have a person responsible for going to the site, analyzing and documenting the data found, the that consumes the employee's work time, which could be performing your time in another activity, as well as it becomes economically not profitable for the company. The main objective of this project is to use the ESP32 controller together with the probe-type temperature sensor to collect information and store it via *WiFi* internet. It also aims to address database storage and subsequent analysis of data collected via *WiFi*, for application in a monitoring system for the industrial maintenance sector. Then elaborate the electronic circuit of the project on the bench and do the proof of concept. And finally, to study the feasibility of applying the low-cost project in an industrial environment.

Keywords: Internet of Things, Temperature sensing, Database

LISTA DE FIGURAS

Figura 1- Esquema de um <i>Chiller</i> resfriado a água – ciclo de refrigeração por compressão com fluido refrigerante genérico.....	15
Figura 2- Arquitetura do ESP32 <i>NodeMCU</i>	20
Figura 3- Sensor de temperatura tipo sonda DS18B20.....	22
Figura 4- Fluxograma de funcionamento do sistema proposto	27
Figura 5- Ilustração esquemática de montagem do sistema em protoboard.....	29
Figura 6- Interface do <i>Firebase Real Time Database</i>	32
Figura 7- Segurança de aplicativos do <i>Real Time Database</i>	33
Figura 8- Teste inicial de bancada do sistema proposto	34
Figura 9- Dados de teste inicial enviados ao banco de dados	35
Figura 10- Externo do painel elétrico com o sistema de coleta de temperatura da água	37
Figura 11- Interno do painel elétrico com o sistema de coleta de temperatura da água	38
Figura 12- Gráfico de temperatura dia 24/10	40
Figura 13- Aquisição de temperatura dia 24/10 com medidor Fluke	41
Figura 14- Gráfico de temperatura dia 17/11	42
Figura 15- Aquisição de temperatura dia 17/11 com medidor Fluke	43

LISTA DE TABELAS

Tabela 1- Leituras de temperatura dia 24/10.....	41
Tabela 2- Leituras de temperatura dia 17/11.....	44

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
EPDM	<i>Ethylene Propylene Diene Methylene</i>
GPIO	<i>General Purpose Input/Output</i>
IDE	<i>Integrated Development Environment</i>
IDF	<i>IoT Development Framework</i>
I ² C	<i>Inter-Integrated Circuit</i>
IoS	<i>Internet of Services</i>
IoT	<i>Internet of Things</i>
JSON	<i>JavaScript Object Notation</i>
MIT	<i>Massachusetts Institute of Technology</i>
NA	Normal Aberto
NF	Normal Fechado
NTP	<i>Network Time Security</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
SDK	<i>Software Development Kit</i>
SPI	<i>Serial Peripheral Interface</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TPV	Termoplástico Vulcanizado
UART	<i>Universal Asynchronous Receiver / Transmitter</i>
ULP	<i>Ultra Low Power</i>
WiFi	<i>Wireless Fidelity</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	MOTIVAÇÃO DO PROJETO	14
1.2	OBJETIVOS	15
1.2.1	Objetivo Geral	16
1.2.2	Objetivos Específicos	16
2	REVISÃO BIBLIOGRÁFICA	17
2.1	EVOLUÇÃO DA INDÚSTRIA	17
2.2	INDÚSTRIA 4.0	18
2.3	IOT NA INDÚSTRIA 4.0	18
2.4	TELEMETRIA	19
2.5	MICROCONTROLADOR ESP32	20
2.6	SENSOR DE TEMPERATURA	21
2.7	BASE DE DADOS EM TEMPO REAL	23
3	ESTRUTURA DO PROJETO	25
3.1	MÉTODO E MATERIAIS	25
3.2	FLUXOGRAMA DO TRABALHO	27
4	RESULTADOS E DISCUSSÃO	28
4.1	DESENVOLVIMENTO DO <i>HARDWARE</i>	28
4.2	DESENVOLVIMENTO DO PROGRAMA	29
4.2.1	Aquisição da temperatura	29
4.2.2	Conexão do ESP32 ao <i>WiFi</i>	30
4.2.3	Programação da data e hora	30
4.2.4	Conexão no banco de dados	31
4.3	DESENVOLVIMENTO DO BANCO DE DADOS	31
4.4	TESTES E SIMULAÇÕES	33
4.5	IMPLEMETAÇÃO NA INDÚSTRIA	36
4.5.1	Local de implementação	36
4.5.2	Materiais para a montagem industrial	36
4.5.3	Inicialização e conexão do dispositivo na rede industrial	38
4.5.4	Experimentação	39
5.	CONCLUSÃO	45

REFERÊNCIAS.....47

1 INTRODUÇÃO

Com a evolução da tecnologia, o mundo passa por uma fase de digitalização e processamento de toda informação para fins analíticos, ou até mesmo, utilizá-la para o desenvolvimento. O próximo passo, é a digitalização constante, de forma rápida e precisa, assim como o compartilhamento dessas informações, a fim de otimizar algum processo. Assim como em uma visão industrial, temos a comunicação entre as máquinas, ou seja, o compartilhamento de dados. Assim surge a *Internet of Things* (IoT) (LOPEZ, 2013).

O conceito de IoT surgiu em 1999 por Kevin Ashton que trabalhava em um laboratório de *AutoID* no *Massachusetts Institute of Technology* (MIT). Kevin pensava sobre máquinas que poderiam trabalhar para as pessoas, coletando dados e comunicando remotamente, interligando todo um sistema. A ideia foi evoluindo com o desenvolvimento da internet, e com o IPv6 que permite a determinação de vários dispositivos em uma rede (LOPEZ, 2013).

Em um ambiente industrial, pode-se dizer que todo processo gera informações, as quais são armazenadas e analisadas, a fim de ter o controle do processo e monitorar com frequência o que acontece em cada processo. Grande parte dos processos industriais ainda são monitorados de forma física, ou seja, é necessário ter um usuário local responsável em ir até o local, analisar e documentar os dados encontrados, o que consome tempo de trabalho do colaborador, que poderia estar desempenhando seu tempo em outra atividade, assim como isso se torna economicamente não rentável para a empresa (SILVA, 2005).

Esse monitoramento se torna rentável, em sistemas remotos, em que é possível saber o que ocorre no processo sem estar no local em que ele acontece. Esse tipo de sistema supervisorio é chamado na literatura de “*Supervisory Control and Data Acquisition*” (SCADA) (SILVA, 2005).

Assim, quando se coleta dados de forma remota, nomeia-se como telemetria, em que se utiliza para medir diversos processos industriais, dependendo da aplicação. É de suma importância a uma indústria ter determinados dados de processos de forma fácil, rápida e intermitente. Dessa forma, evita-se problemas maiores, como por exemplo, se a água industrial não está na temperatura correta na qual o processo de produção demanda, todas as peças produzidas têm sua qualidade prejudicada, gerando um prejuízo para a empresa. Por isso, é de extrema importância, o

departamento de manutenção industrial acompanhar esse tipo de dado de forma constante (NEOWATER, 2021).

Entende-se então, que a evolução do IoT está diretamente relacionada ao fato que sistemas inteligentes e redes se desenvolvem e evoluem cada dia mais, ficando cada vez mais fácil a interface de comunicação entre máquinas e também com redes. O principal fator que auxilia a elaboração de projetos IoT, é a praticidade em implementação em rede de alguns microcontroladores, pois são de fácil integração em rede de *Transmission Control Protocol/Internet Protocol* (TCP/IP), além de contar com *Wireless Fidelity* (WiFi) (BABIUCH, 2019).

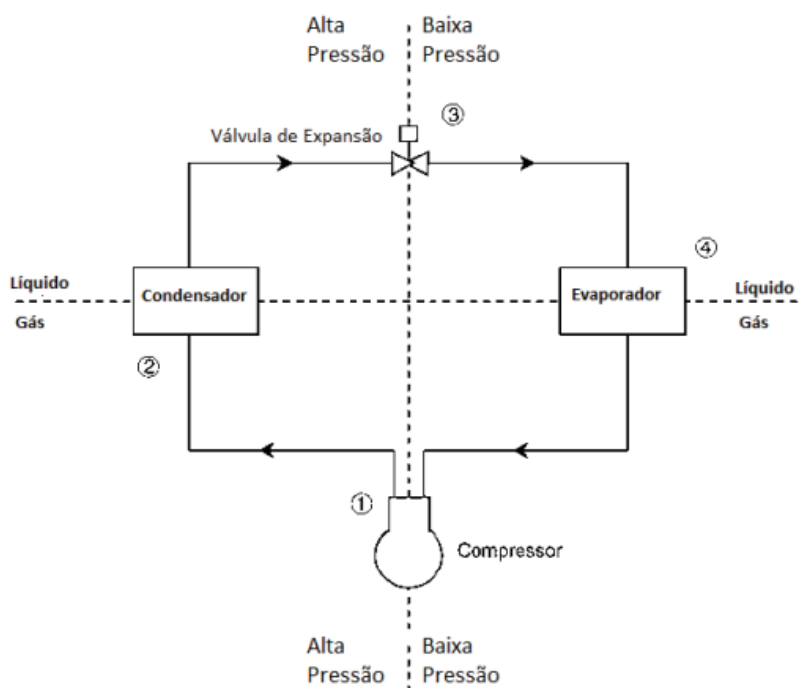
Contudo, nesse trabalho, para realizar a aquisição e possível exportação dos dados, é necessário utilizar um banco de dados que armazenará todos os dados lidos. O *Firebase* é uma plataforma de desenvolvimento do Google que pode realizar a análise, autenticação, bancos de dados, configuração, armazenamento de arquivos, envio de mensagens entre outras funções. Logo, torna-se viável a utilização, pois sua utilização em conjunto com o ESP32, permite a comunicação com diferentes aparelhos conectados à rede, de forma simultânea e em tempo real, assim fica mais fácil realizar a exportação dos dados (FIREBASE, 2023).

1.1 MOTIVAÇÃO DO PROJETO

Em uma indústria, encontra-se alguns processos que necessitam de circulação de água para o resfriamento, por esse motivo, é necessário a utilização de geladeiras industriais ou *chillers*, que tem a funcionalidade de manter a temperatura da água baixa o suficiente, a fim de contemplar o processo.

Em algumas empresas utiliza-se os *chillers* resfriados à água, com sistema de expansão indireta. De acordo com *Stanford* (2003), o funcionamento deste equipamento consiste em um fluido refrigerante que sofre compressão no compressor, e assim é enviado para o condensador, onde o fluido refrigerante realiza a troca térmica com a água de condensação (água da torre industrial), resfriando o fluido refrigerante. Após o condensador, o fluido passa por um processo de expansão, na válvula de expansão, que logo em seguida segue para o evaporador. No evaporador, o fluido de refrigeração troca calor com a água de resfriamento (água utilizada no processo industrial), resfriando-a. O processo continua constante em um ciclo fechado, como observa-se na Figura 1.

Figura 1- Esquema de um *Chiller* resfriado a água – ciclo de refrigeração por compressão com fluido refrigerante genérico



Fonte: Adaptado de Stanford (2003).

Verifica-se que o bom funcionamento do *chiller*, assim como garantir a qualidade da produção de peças, é necessário realizar o monitoramento e análise da temperatura da água em todos os estágios do processo, seja na água de condensação, assim como na água de resfriamento.

Esse processo de coleta de temperatura da água sempre foi feito de forma visual na indústria estudada, e com sensores termopar locais, o que requer que um colaborador vá até o local de inspeção e anote manualmente a temperatura.

A motivação desse projeto é utilizar o conceito de IoT para realizar essa coleta de temperatura, permitindo a fácil análise e monitoramento do processo sem o deslocamento físico. Logo, salva-se tempo de trabalho aos colaboradores da empresa, que podem realizar outra atividade.

1.2 OBJETIVOS

O presente trabalho tem como objetivo principal a implementação de um sistema de monitoramento da temperatura da água do processo industrial, e que melhor se adeque ao uso do departamento de manutenção industrial, implementando o conceito de IoT e automatização do processo de leitura. Em complemento, fazer um

estudo sobre a indústria e o conceito de IoT, telemetria utilizando microcontrolador e a aplicação desse conceito na indústria.

1.2.1 Objetivo Geral

Propor uma metodologia automática de coleta da temperatura da água industrial e implementar o conceito de IoT com microcontrolador na indústria.

1.2.2 Objetivos Específicos

- Estudar a evolução da indústria
- Estudar o conceito de IoT na indústria 4.0
- Estudar o conceito de telemetria
- Analisar o microcontrolador e o sensor de temperatura utilizado
- Analisar os sistemas de banco de dados que receberá os dados de temperatura
 - Aplicação do sistema em ambiente industrial, estudando a viabilidade e adequando ao uso do departamento de manutenção industrial.

2 REVISÃO BIBLIOGRÁFICA

Com a finalidade de entender os caminhos que a indústria tomou nos últimos tempos, assim como sua evolução, estudou-se o avanço da tecnologia aplicada, até a indústria atual, além de alguns equipamentos empregados do IoT.

2.1 EVOLUÇÃO DA INDÚSTRIA

A indústria tem passado por várias transformações ao longo da história, impulsionadas pelo avanço da tecnologia e pela busca por maior eficiência e produtividade. Portanto, o avanço se deu em várias revoluções industriais. As três primeiras revoluções tiveram papel fundamental para que houvesse uma quarta revolução, o surgimento da indústria 4.0 (SAKURAI, 2018).

De acordo com Venturelli (apud SAKURAI, 2018, p. 482), a primeira revolução industrial ocorreu no final do século XVIII e início do século XIX, com a introdução das máquinas a vapor, mecanização da produção e também uma evolução no setor de transporte. Isso trouxe a substituição do trabalho manual pelo trabalho mecanizado, resultando em um aumento significativo na produtividade e na capacidade de produção, além da descoberta pela ciência da fonte de energia proveniente do carvão.

A segunda revolução industrial segundo Boettcher (apud SAKURAI, 2018, p. 483), ocorreu no final do século XIX e início do século XX, impulsionada pela eletrificação e novas tecnologias. Além disso, a introdução das linhas de produção em massa, como o famoso modelo de produção de Henry Ford, permitiu a produção em larga escala e a padronização dos produtos.

A terceira revolução industrial, também conhecida como revolução Técnico-Científica e Informacional segundo Boettcher (apud SAKURAI, 2018, p. 484), teve início na segunda metade do século XX com o advento da eletrônica, da informática e das telecomunicações. A utilização de máquinas se tornou cada vez mais avançada com a utilização de sistemas de controle computadorizados e robôs industriais. A tecnologia da informação desempenhou um papel crucial, além da utilização de várias fontes de energia. Surge também a nanotecnologia, avanços nos meios de comunicação e também na biotecnologia. Também é importante citar nessa fase, a globalização. De acordo com SILVA *et. al.* (apud SAKURAI, 2018, p. 484), surgem-se

as potências industriais com utilização massiva de produtos tecnológicos, assim como ressalta-se nessa fase a importância dada à consciência ambiental.

2.2 INDÚSTRIA 4.0

Segundo Silveira (apud SAKURAI, 2018, p. 485), existem seis princípios fundamentais para a aplicação da indústria 4.0. Em resumo, são eles:

- Aquisição e tratamento de dados em tempo real.
- Virtualização do ambiente industrial.
- Descentralização com a utilização de um sistema cyber-físico para tomada de decisões.
- Atrelar IoT à *softwares*, conceito chamado de *Internet of Services (IoS)*.
- Flexibilidade em modularidade da indústria, podendo acoplar e desacoplar equipamentos de acordo com a produção.
- Conexão entre máquinas criando um conceito de fábrica inteligente.

Além disso, Sakurai destaca as principais tecnologias que surgiram com a indústria 4.0. Sendo, a internet das coisas, segurança cibernética, big data, computação em nuvem, robótica avançada, inteligência artificial, e novos materiais de propriedades “inteligentes” (SAKURAI, 2018).

Nesse trabalho destaca-se o uso da internet das coisas.

2.3 IOT NA INDÚSTRIA 4.0

Segundo SCHWAB (apud SAKURAI, 2018, p. 486), a Internet das Coisas é “a relação entre coisas (produtos, serviços, lugares) e pessoas, por meio de plataformas e tecnologias conectadas”.

O IoT permite a criação de fábricas inteligentes, onde os dispositivos estão interconectados e podem se comunicar uns com os outros. Isso possibilita a automação e a coordenação eficiente de processos industriais complexos. Por exemplo, máquinas podem ajustar automaticamente suas configurações com base em dados recebidos de outras máquinas ou sistemas. Isso aumenta a flexibilidade e a adaptabilidade da produção, permitindo a personalização em massa e a nas demandas do mercado (FEREIRA, I, V. et al).

Outro fator de caracteriza o IoT, é a ampliação da análise de dados avançada na indústria 4.0. Com a enorme quantidade de dados gerados pelos dispositivos conectados, é possível aplicar técnicas de análise de big data, inteligência artificial e aprendizado de máquina para extrair insights valiosos. Os algoritmos podem identificar padrões, prever falhas, otimizar processos e tomar decisões autônomas com base nos dados coletados. Isso leva resposta rápida às mudanças a uma produção mais eficiente, redução de custos e melhoria da qualidade dos produtos (FEREIRA, I, V. et al).

2.4 TELEMETRIA

A telemetria pode ser descrita como uma tecnologia pela qual todas as coisas mensuráveis podem ser controladas, além de visualizar os dados do dispositivo, como hora e localização, é uma técnica completa para obter relatórios precisos. A tecnologia fornece informações diferenciadas sobre algumas funções, exemplos mais complexos de uso das soluções adotadas e medição de temperatura e pressão (BRANCO apud FONSECA, 2012, p. 40).

Mattos afirma que a tecnologia de telemetria surgiu com a necessidade de realizar e avaliar medições que precisavam ser feitas em locais de difícil acesso, para isso era necessário um sistema que observasse esses pontos inacessíveis. Reforçando o conceito, Vissotto Junior (2004) e Santos (2010), dizem que a tecnologia permite o fornecimento de conectividade entre os subsistemas envolvidos, pois seu objetivo não é apenas fornecer serviços aos usuários, mas também melhorar o gerenciamento e a segurança do processamento da informação, bem como ajudar as organizações a entender melhor seus próprios mercados como seus clientes. Além de uma comunicação harmoniosa entre os departamentos, são fornecidos melhores produtos e serviços (apud FONSECA, 2012, p. 41).

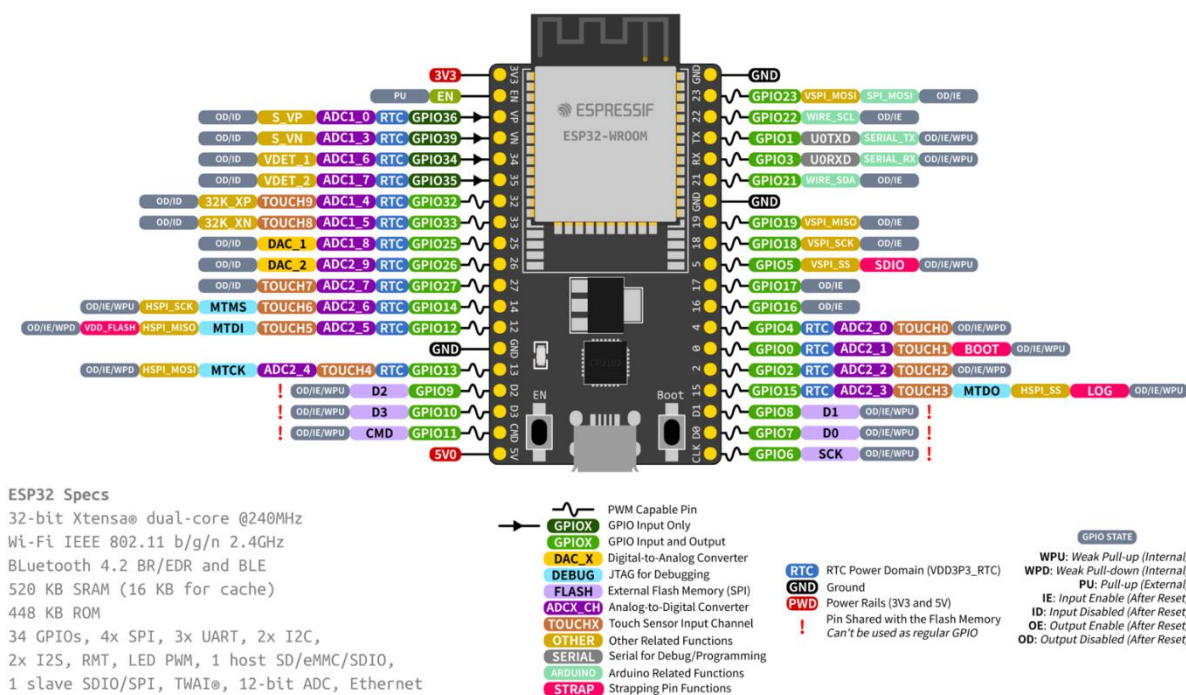
Nesse sentido, fica evidente que a IoT presente na indústria 4.0 ajuda a evolução de sistemas de telemetria conectando processos e rede para captação de dados a qualquer momento em qualquer lugar.

2.5 MICROCONTROLADOR ESP32

Pode-se dizer que o ESP32 é um microcontrolador de baixo custo e alto desempenho que se tornou extremamente popular na comunidade de eletrônicos e desenvolvimento de projetos de Internet das Coisas (IoT). Lançado pela *Espressif Systems*, o ESP32 é uma evolução do seu antecessor, o ESP8266, e oferece uma variedade de recursos e funcionalidades que o tornam uma escolha poderosa para uma ampla gama de aplicações, que inclui a indústria. A Figura 2 apresenta a arquitetura do microcontrolador.

Figura 2- Arquitetura do ESP32 NodeMCU

ESP32-DevKitC



Fonte: (ESPRESSIF, 2023)

De acordo com *ESPRESSIF*, o microcontrolador tem a capacidade de se conectar a redes *WiFi* e *Bluetooth* simultaneamente, o que o torna ideal para projetos de IoT que requerem conectividade sem fio. Com uma antena embutida, ele permite a comunicação com outros dispositivos, como *smartphones*, *tablets* e servidores em nuvem, facilitando a troca de dados e o controle remoto. A facilidade em trabalhar com

o microcontrolador sem fio o torna muito atrativo no ambiente industrial, onde há uma grande quantidade de cabos e fios.

Além disso, o ESP32 possui um processador de 32 bits, o que o torna capaz de lidar com tarefas mais complexas e processamento paralelo. Isso é especialmente útil para projetos que exigem multitarefa ou o processamento rápido de dados, como aplicações de monitoramento (*ESPRESSIF*, 2023).

O microcontrolador ESP32 também oferece uma quantidade generosa de memória, incluindo memória *RAM* e flash, permitindo que você armazene e execute programas mais extensos e utilize bibliotecas adicionais. Ele suporta várias interfaces, como *General Purpose Input/Output* (GPIOs), *Universal Asynchronous Receiver / Transmitter* (UART), *Serial Peripheral Interface* (SPI) e *Inter-Integrated Circuit* (I²C), o que facilita a conexão com sensores, atuadores e outros dispositivos externos (*ESPRESSIF*, 2023).

Uma característica interessante do ESP32 que a *ESPRESSIF* apresenta é a presença de um coprocessador de baixa potência, chamado de *Ultra Low Power* (ULP), que permite que o microcontrolador execute tarefas simples em um modo de consumo de energia extremamente baixo. Isso é particularmente útil em projetos de IoT que exigem longa duração da bateria, como dispositivos de monitoramento remoto ou sensores de baixa potência.

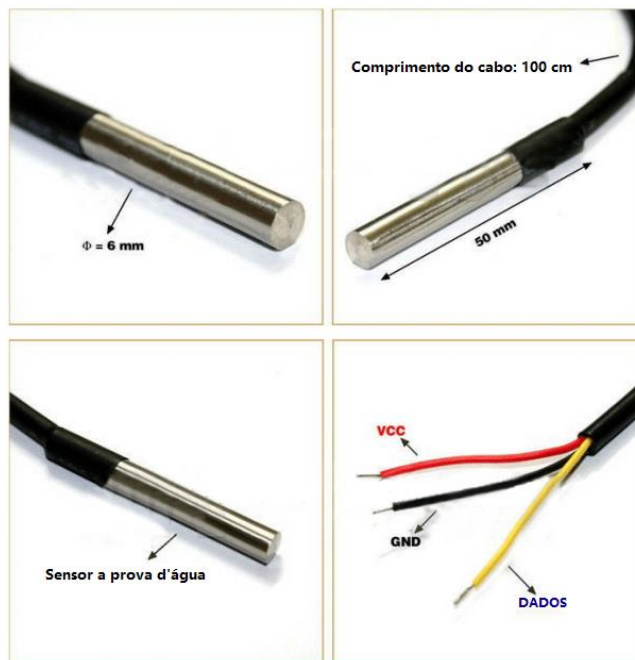
Além de suas boas especificações técnicas, o ESP32 também é suportado por uma comunidade ativa de desenvolvedores. Isso significa que existem muitos recursos, tutoriais e bibliotecas disponíveis, o que facilita a criação de projetos e a resolução de problemas. Além disso, a *Espressif Systems* oferece um ambiente de desenvolvimento integrado (IDE) chamado ESP-IDF (*ESP32 IoT Development Framework*), que fornece uma base sólida para programação e depuração de projetos no ESP32. Também é possível utilizar o ambiente de desenvolvimento do famoso Arduino, chamado Arduino IDE, contudo, deve-se incluir algumas bibliotecas de trabalho (*ESPRESSIF*, 2023).

2.6 SENSOR DE TEMPERATURA

O sensor de temperatura DS18B20 é um dispositivo eletrônico amplamente utilizado para medir a temperatura de maneira precisa e confiável. O DS18B20 é um sensor digital de temperatura de alta precisão tipo sonda como é possível ver na

Figura 3, com uma interface que pode ser utilizada com microcontroladores de baixa tensão (ELETROGATE, 2023).

Figura 3- Sensor de temperatura tipo sonda DS18B20



Fonte: (ELETROGATE, 2023)

Uma das principais características do DS18B20 é sua capacidade de operar em uma ampla faixa de temperatura, que varia de -55°C a $+125^{\circ}\text{C}$. Isso o torna adequado para uma variedade de aplicações, desde monitoramento de temperatura em ambientes industriais até medição de temperatura em equipamentos domésticos (*DS18B20 DataSheet*).

O DS18B20 utiliza o princípio de medição de temperatura baseado na variação da resistência elétrica de um sensor de platina. Esse sensor é encapsulado em um invólucro à prova d'água, tornando-o adequado para uso em ambientes externos ou úmidos. O encapsulamento também garante uma resposta térmica rápida e precisa, permitindo que o sensor forneça leituras confiáveis em tempo real (ELETROGATE).

A interface de comunicação digital do DS18B20 é feita através de um fio (protocolo *ONEWIRE*), o que torna de fácil acesso a microcontroladores ou outros dispositivos eletrônicos com apenas um fio de dados. Além disso, vários sensores DS18B20 podem ser conectados em paralelo ao mesmo fio de dados, simplificando a leitura de várias temperaturas em um único sistema (*DS18B20 DataSheet*).

Sobre a precisão do DS18B20, ele possui uma resolução de 12 bits, o que significa que pode fornecer leituras com uma precisão de até $0,0625^{\circ}\text{C}$. Isso é

extremamente útil em aplicações que exigem medições precisas e confiáveis, como sistemas de controle de temperatura, dispositivos médicos ou experimentos científicos (DS18B20 *DataSheet*).

Além disso, o DS18B20 é um sensor de baixo consumo de energia, o que o torna adequado para aplicações alimentadas por baterias ou outros sistemas com restrições energéticas. Ele também possui recursos de conversão automática de temperatura e detecção de presença, o que o torna eficiente em termos de recursos do sistema (DS18B20 *DataSheet*).

Em resumo, o sensor de temperatura DS18B20 é uma escolha popular devido à sua alta precisão, facilidade de uso, ampla faixa de temperatura e capacidade de operar com baixo consumo de energia. Seja para monitorar a temperatura em um ambiente industrial, controlar a temperatura de um sistema doméstico ou realizar experimentos científicos, o DS18B20 se destaca como uma sonda confiável e precisa para medição de temperatura (*ELETROGATE*).

2.7 BASE DE DADOS EM TEMPO REAL

O *Firebase* Realtime Database é um serviço de banco de dados em tempo real oferecido pelo Google. Ele permite que os desenvolvedores criem aplicativos com recursos de sincronização em tempo real, onde os dados são atualizados e refletidos instantaneamente em todos os dispositivos conectados (*FIREBASE*, 2023).

O banco de dados é baseado em uma estrutura de árvore de dados hierárquica de formato JSON. Os dados são armazenados como pares de chave-valor, onde cada nó na árvore de dados tem uma chave única. O desenvolvedor pode escrever e ler eventos em qualquer nó da árvore, permitindo que eles personalizem a lógica do aplicativo com base nas alterações nos dados (*FIREBASE*, 2023).

Para acessar o *Firebase* Realtime Database, os desenvolvedores podem usar as bibliotecas de *software* do *Firebase* disponíveis para várias plataformas, como Android, iOS, Web e várias outras. Essas bibliotecas fornecem uma API simples e intuitiva para interagir com o banco de dados, permitindo que os desenvolvedores se concentrem na lógica do aplicativo em vez de se preocupar com detalhes de implementação (*FIREBASE*, 2023).

O *Firebase* Realtime Database pode ser uma solução prática e eficiente para integrar dispositivos de IoT aos seus aplicativos, fornecendo sincronização em tempo

real, armazenamento de dados e recursos de automação. Combinado com as outras ferramentas do *Firebase*, como autenticação de usuários e notificações push, você pode criar uma experiência completa e interativa para os usuários que utilizam dispositivos de IoT (*FIREBASE*, 2023).

3 ESTRUTURA DO PROJETO

Para a realização do trabalho, foi necessário seguir uma sequência de passos, assim como testes a partir de um protótipo criado. Assim, estruturou-se o projeto em duas etapas.

3.1 MÉTODO E MATERIAIS

Os materiais utilizados para o projeto de telemetria IoT foram um módulo ESP32 – *Node MCU*; uma fonte de alimentação com entrada 220 Vac e saída 24 Vdc de baixa potência, sendo que não é necessário a utilização de grandes potências para utilizar o microcontrolador e sensor de temperatura, porém, é necessário uma fonte utilizada em aplicações industriais; um abaixador de tensão ajustável que converterá os 24 Vdc em 5 Vdc e 3,3 Vdc que é a tensão utilizada no ESP32; uma protoboard; jumper; sensor de temperatura tipo sonda DS18B20; computador e cabo USB - Micro USB; além de um painel para realizar a aplicação na indústria.

Inicialmente, fez-se revisão bibliográfica sobre a indústria 4.0 e as aplicações industriais IoT com a evolução das tecnologias e microcontroladores. Por se tratar de uma estrutura de baixo custo, optou-se em utilizar o microcontrolador ESP32- Node MCU. Realizou-se uma pesquisa e estudo baseado nesse microcontrolador em questão, a fim de entender suas entradas e saídas, funcionalidades e tensão de operação.

Após feito o estudo e decidido a estrutura de ligação para o funcionamento do ESP32 em conjunto com o sensor de temperatura, foi possível iniciar a programação do microcontrolador com o auxílio do *software* Arduino IDE, que a princípio é utilizado para a programação do microcontrolador Arduino, mas expansível para o ESP32 quando utiliza-se de plug-ins internos ao *software*.

Programou-se o microcontrolador para realizar a aquisição da temperatura instantânea, a qual é captada pelo sensor, e também a conexão do ESP32 à rede sem fio (*WiFi*). Também se viu necessário a captação da data e hora no momento em que é realizada a leitura, para que posteriormente seja identificado no banco de dados.

Após feito a programação inicial do ESP32, foi necessária a criação de uma estrutura de banco de dados gratuita e que seja capaz de gravar todos os dados lidos pela sonda de temperatura DS18B20. Para isso, utilizou-se o banco de dados

proveniente da empresa Google, o *Firebase Real Time Database*. Foi necessário realizar uma programação utilizando o mesmo programa do *software* Arduino IDE, destinada ao vínculo com o banco de dados e a criação de uma estrutura própria para o projeto.

Para realizar a consulta dos dados disponíveis no *Firebase Real Time Database*, foi possível utilizar um *software* de fácil acesso, e que realiza a integração ao banco de dados, o *Power Query*, o qual é integrado ao Excel. O *Power Query* é responsável por realizar as consultas referente a cada data que o microcontrolador insere no banco de dados, logo, ele busca o arquivo JSON do *Firebase* diretamente no diretório do site, e atualiza as informações consultadas em qualquer momento desejado.

É possível importar a consulta do *Power Query* para o Excel, que dará os dados do *Firebase* em forma de tabela para uma planilha no Excel, podendo até gerar gráficos. Tudo isso, torna o projeto simples, pois estamos utilizando o Excel, que se configura como uma ferramenta de uso constante e rotineiro pelos colaboradores de uma empresa.

Com a estrutura de *hardware*, programação, banco de dados e consulta finalizados, fez-se a experimentação do circuito em bancada. Nessa etapa do projeto, já foi possível realizar a montagem do sistema em um painel elétrico, assim como a utilização da fonte 24 Vdc que tem seu uso comum na indústria. A fonte de corrente contínua industrial é mais robusta que uma fonte comum, e pode suportar oscilações, defasagem e até minimamente uma ondulação de corrente. Além disso, para a alimentação do microcontrolador, foi necessário um circuito abaixador de tensão (*StepDown*), que regula a tensão até a necessária.

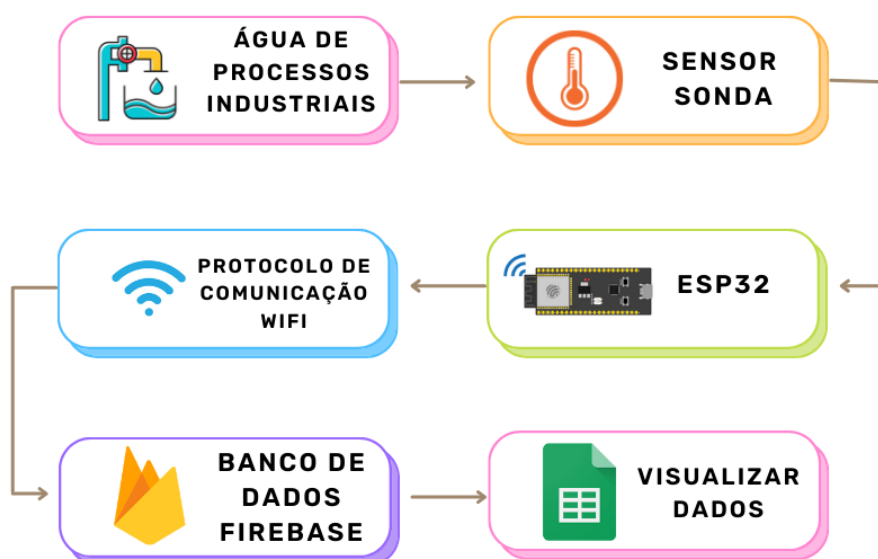
Por fim, instalou-se o painel elétrico no ambiente industrial com o intuito de colocar à prova o equipamento e realizar os testes. Outro desafio nessa etapa foi conectar o dispositivo na rede sem fio *WiFi* industrial, o que foi realizado com o apoio da equipe de TI. Utilizou-se um *WiFi* próprio para aplicações industriais, o qual foi desenvolvido com uma segurança específica para uso em máquinas e que diferencia da rede de uso em computadores de escritório.

Em consequência, o equipamento foi testado e monitorado. Experimentou-o em ambiente industrial, onde se tem interferências físicas, como sujeira, calor, assim como problemas de variação da rede elétrica industrial.

3.2 FLUXOGRAMA DO TRABALHO

O sistema proposto visa realizar o transporte da informação, a qual é a temperatura da água industrial, que com o modelo de IoT, irá levar esse dado de forma rápida e automática até o observador. Assim, essa informação possui um caminho a ser seguido, que também foi um caminho para a construção do projeto. A Figura 4 representa o fluxograma onde toda a informação captada circula.

Figura 4- Fluxograma de funcionamento do sistema proposto



Fonte: Próprio autor.

É possível observar na Figura 4 todo o processo de aquisição de dados, onde a informação será captada inicialmente pelo sensor tipo sonda imergido na água do processo industrial, que após ser interpretada pelo microcontrolador, utilizará do protocolo de rede em comunicação sem fio (*WiFi*) para arquivar a temperatura em banco de dados. Por fim, pode-se consultar e visualizar os dados através do *Power Query* e Excel.

4 RESULTADOS E DISCUSSÃO

Ao início da construção do sistema foi necessário seguir uma etapa de elaboração, pois cada passo depende de um desenvolvimento posterior.

Com isso, iniciou-se os trabalhos de desenvolvimento de *hardware*, verificando a melhor ligação para conexão dos componentes. Após isso, foi necessário acumular os dados captados em um banco de dados, que por fim seriam visualizados.

4.1 DESENVOLVIMENTO DO *HARDWARE*

A princípio, foi preciso realizar a ligação do módulo ESP32 *NodeMCU* com o sensor de temperatura DS18B20 tipo sonda. Assim como se viu na revisão bibliográfica do sensor, ele possui 3 fios, sendo o preto GND, o vermelho, a alimentação 5 Vdc ou 3,3 Vdc proveniente do ESP32, e também o fio amarelo, o qual transmite os dados do sensor para uma porta digital do ESP32.

Assim, ligou-se o fio vermelho do sensor na porta 3V3, o fio preto na porta GND e o amarelo, na entrada digital TX2 ou GIOP17.

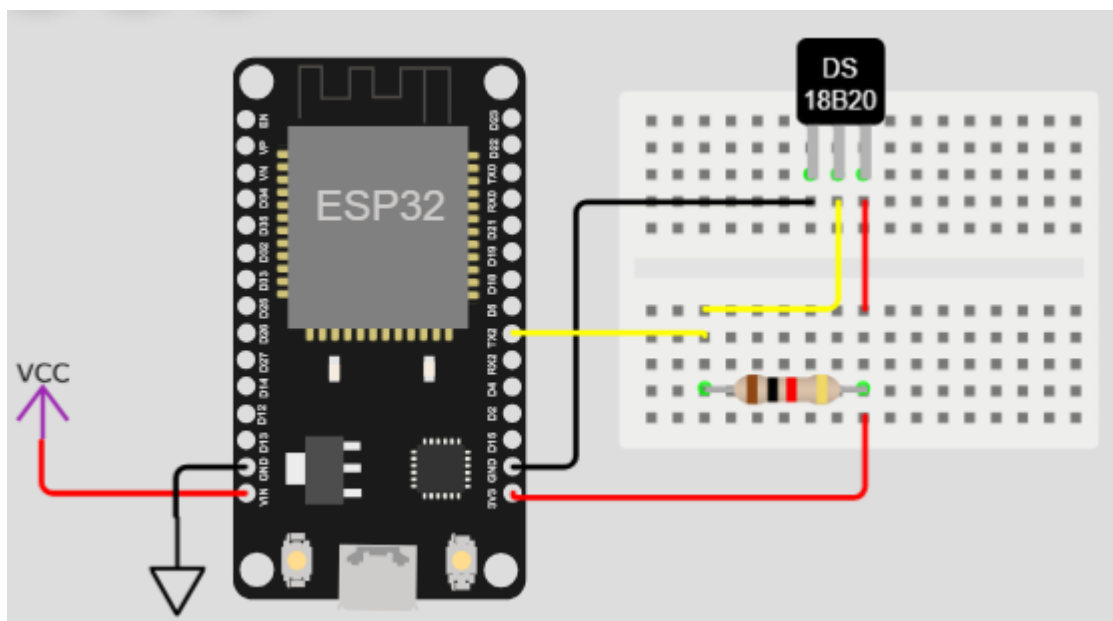
A fim de garantir que o sinal da GIOP17 permaneça em um valor conhecido enquanto não recebe nenhuma tensão, instalou-se um resistor de *pull-up*. Esse resistor, garante que o sinal lido seja de nível alto enquanto não é realizado a leitura do sensor, assim a leitura do pino digital não ficará flutuando.

Deve-se garantir que a corrente que circula pelo sensor, seja menor que 1mA. Assim, utilizou-se de um sensor de 4,7kΩ que atende com perfeição.

$$I = \frac{V}{R} = \frac{3,3Vdc}{4,7k\Omega} = 0,7mA$$

A ilustração do circuito da Figura 5, mostra como foi feita a ligação em protoboard, assim como apontado. O *software* utilizado para tal ilustração não possuía o sensor de temperatura DS18B20 tipo sonda, portanto, foi utilizado o sensor DS18B20 comum, no entanto, tanto o tipo sonda, quanto o comum possuem as mesmas saídas e características, a diferença é apenas que o sensor tipo sonda tem a capacidade de operar a prova d'água.

Figura 5- Ilustração esquemática de montagem do sistema em protoboard



Fonte: Próprio autor.

Observa-se na Figura 5, que a alimentação do sensor tipo sonda é ligado no pino 3,3 Vdc, e o pino que realiza a leitura de dados também é alimentado através do resistor.

4.2 DESENVOLVIMENTO DO PROGRAMA

A fim de programar o microcontrolador, a ferramenta utilizada foi o ArduinoIDE, uma poderosa ferramenta, que pode programar o ESP32 quando devidamente configurada. É possível ver o programa criado no Apêndice A.

4.2.1 Aquisição da temperatura

Inicialmente no desenvolvimento do programa, procurou-se realizar o que seria sua função principal em conjunto ao ESP32, que é a comunicação com o sensor de temperatura tipo sonda, e a leitura da temperatura através dele.

Para isso, foi necessário a inclusão da biblioteca “*OneWire.h*”. A biblioteca *OneWire* é uma ferramenta de código aberto usada em sistemas embarcados para permitir a comunicação de dados através de um único fio de conexão, sendo amplamente empregada em interfaces de sensores e dispositivos de baixo custo em

aplicações de eletrônica e automação. Logo, é necessário no caso do sensor DS18B20, pois sua comunicação se dá apenas pelo fio de cor amarelo.

Além disso, com a utilização desse tipo de sensor (de padrão de comunicação *OneWire*), torna-se necessário a inclusão da biblioteca “DallasTemperature.h”. Essa biblioteca possibilita a comunicação e controle de sensores de protocolo *OneWire* desenvolvidos pela Dallas Semiconductors.

Logo, é possível realizar a leitura a temperatura em graus Celsius, proveniente do sensor tipo sonda DS18B20, toda a programação realizada, é possível ver no anexo I.

4.2.2 Conexão do ESP32 ao *WiFi*

Um dos motivos da escolha do microcontrolador ESP32, foi a sua integrada e fácil conexão a rede sem fio (*WiFi*), o que facilitará todo o processo.

Assim, foi necessário a utilização a biblioteca “*WiFiUdp.h*”. É uma ferramenta baseada na plataforma Arduino, porém também disponível ao ESP32, e oferece recursos para a comunicação eficiente e confiável por meio de redes *WiFi* utilizando o protocolo UDP (*User Datagram Protocol*). Considerando que em aplicações industriais pode acontecer a perda ocasional de pacotes de dados, o protocolo UDP realiza isso sem prejudicar a integridade das operações, tornando a comunicação mais robusta em ambientes industriais, que são lugares voláteis em questões de frequências e rede elétrica.

4.2.3 Programação da data e hora

Portanto, tendo a programação para aquisição de temperatura em mãos e também a conexão ao *WiFi*, basta agora enviar tais dados para um local que irá armazenar todas as medições feitas pelo sensor sonda. Porém, observou-se que as medições iriam ficar arbitrárias no banco de dados, e assim não se teria quando foi feita a leitura. Assim, veio a necessidade de registrar a data e a hora no instante em que a medição foi realizada. Ou seja, na repetição da programação, a cada ciclo de programa, o mesmo deve informar que dia e que horas foi feita a medição instantânea.

Por conseguinte, utilizou-se a biblioteca “*TimeLib.h*”. Essa biblioteca é usada para lidar com dados temporais, oferecendo funções para representar, converter e

manipular datas, horários e cronometragem. Ela possibilita a sincronização com fontes precisas de tempo e facilita o tratamento de fusos horários, logo, é muito útil nesse tipo de projeto, o qual exige rastreamento temporal preciso para o controle do processo.

Outro fator que torna ainda mais preciso a obtenção do horário instantâneo de leitura do sensor sonda é a biblioteca "*NTPClient.h*". Tal ferramenta sincroniza relógios internos de dispositivos conectados à Internet com servidores de *Network Time Security* (NTP). Ou seja, facilitando a obtenção precisa da data e hora atual, o que é essencial para esse tipo de projeto. Além de sua utilização simplificar e atualizar constantemente a comunicação com os servidores NTP.

4.2.4 Conexão no banco de dados

A fim de sincronização do ESP32 com o *Real Time Database* do *Firebase*, inicialmente foi necessário a instalação da biblioteca "*FirebaseESP32.h*". Desenvolvida para a plataforma ESP32, a biblioteca "*FirebaseESP32.h*" é uma ferramenta de *software*, que permite a integração fácil do dispositivo com o serviço *Firebase* do Google. Logo, permite-se a utilização do ESP32 com o *Real Time Database*, que possibilita a realização desse tipo de projeto IoT.

O código utilizado no programa para enviar as informações de temperatura ao *Firebase* é "*Firebase.setFloat (fbdo, nodePath + "/temperatura", temperatura);*". Ou seja, envia os dados de valor *Float* para o banco de dados. Assim, ele envia os dados contidos da variável *Float* "temperatura" direto para o nó da árvore de dados escolhida do *Real Time Database*.

Assim, utilizando todas as credenciais de segurança do banco de dados necessárias e conexão à rede sem fio, é possível vincular o ESP32 ao *Firebase*.

O programa final está disposto no Apêndice A.

4.3 DESENVOLVIMENTO DO BANCO DE DADOS

Com o decorrer do desenvolvimento do programa, foi possível fazer a gravação da temperatura instantânea registrada pelo sensor em uma variável de temperatura criada na programação realizada do Arduino IDE. Portanto, a cada momento em que o microcontrolador realizasse uma leitura, precisou-se registrar o valor contido nessa

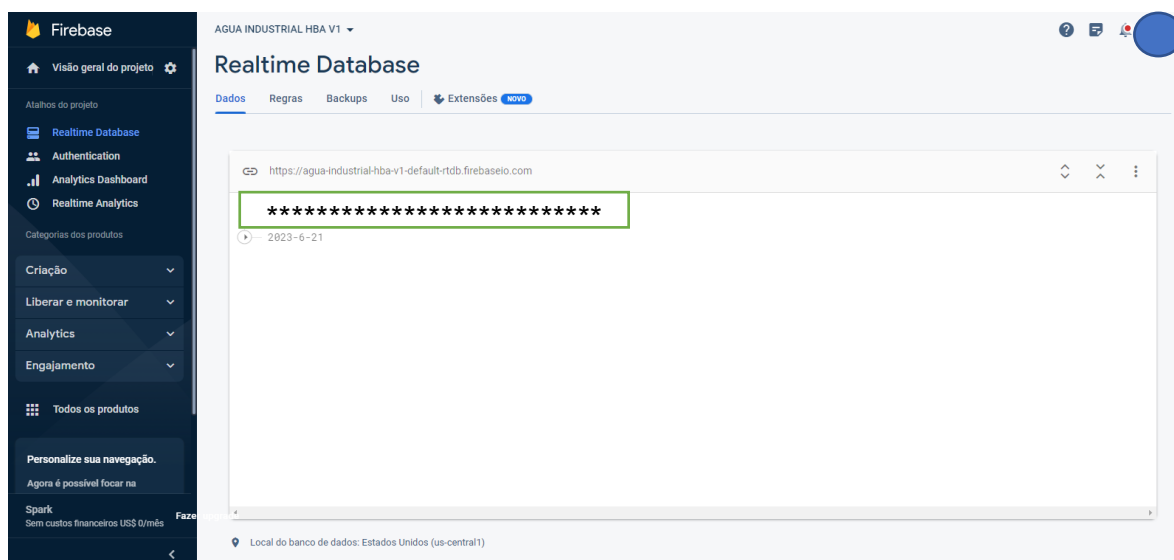
variável em um banco de dados. Para isso, escolheu-se o *Firebase Realtime Database*.

Assim como visto em revisão bibliográfica, o banco de dados é capaz de acompanhar a programação realizada no ESP32 baseado em uma estrutura de árvore de dados hierárquica de formato JSON, em que os dados são armazenados como pares de chave-valor.

A princípio, foi necessário criar uma conta no Google *Firebase* a partir de uma conta google, em que permite a criação gratuita de um banco de dados. Na sequência, pode-se criar um banco de dados em tempo real.

A Figura 6 apresenta como é a visualização da tela do *Real Time Database*.

Figura 6- Interface do *Firebase Real Time Database*



Fonte: Próprio autor

Como é possível observar na Figura 6, o nome dado o banco de dados é “ÁGUA INDUSTRIAL HBA V1”. O *Firebase* cria uma estrutura inicial para que posteriormente se faça as alterações necessárias, assim como a manipulação da árvore de dados.

Para extrair dados da árvore de dados do *Firebase*, foi necessário realizar o vínculo entre o *Power Query* do Excel, com o *Real Time Database*, o *Power Query* é capaz de realizar consultas diretamente no banco de dados, extraíndo o arquivo JSON de forma online. Para isso, o *Firebase* oferece um arquivo com as informações e chaves privadas do banco de dados, o que o torna seguro. Essa segurança do *Firebase* é chamada *Firebase Authentication* (SDK), e pode ser usada para integrar

manualmente um ou mais métodos de login em um aplicativo. Ou seja, foi necessário fornecer os dados necessários ao *Power Query* do Excel, para que o *Firebase* forneça o vínculo. É possível ver as informações de SDK mostradas pelo *Firebase* na Figura 7.

Figura 7- Segurança de aplicativos do *Real Time Database*

Seus aplicativos

Adicionar app

Apps da Web

Água industrial HBA
App da Web

Apelido do app
Água industrial HBA

ID do aplicativo

Vincular a um site do Firebase Hosting

Configuração do SDK

NPM CDN Config

Accesse o snippet do seu objeto de configuração do Firebase do app. [Saiba mais](#).

Objeto de configuração do Firebase que contém as chaves e identificadores do seu app:

```
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "*****",
  authDomain: "*****",
  databaseURL: "*****",
  projectId: "*****",
  storageBucket: "*****",
  messagingSenderId: "*****",
  appId: "*****",
  measurementId: "*****"
};
```

Você está usando o npm e um bundler como Webpack ou Rollup? Consulte o [SDK modular](#).

Saiba mais sobre o Firebase para Web: [Vamos começar](#), [Referência da API Web SDK](#), [Amostras](#)

Fonte: Próprio autor

4.4 TESTES E SIMULAÇÕES

Com o material, montagem do *hardware* e *software* desenvolvidos, foi possível realizar um estudo a respeito da implementação do sistema. Ou seja, iniciou-se a fase de testes de bancada do dispositivo.

Esse tipo de teste foi realizado a fim de prever possíveis falhas durante o processo, e também para verificar o funcionamento pleno do dispositivo, tendo em vista que tudo será feito em ambiente industrial, onde a rede elétrica é saturada de motores e máquinas que prejudicam a qualidade da energia elétrica.

A Figura 8 apresenta como foi realizado o primeiro teste de bancada com o circuito e também realizado algumas leituras de temperatura contínuas.

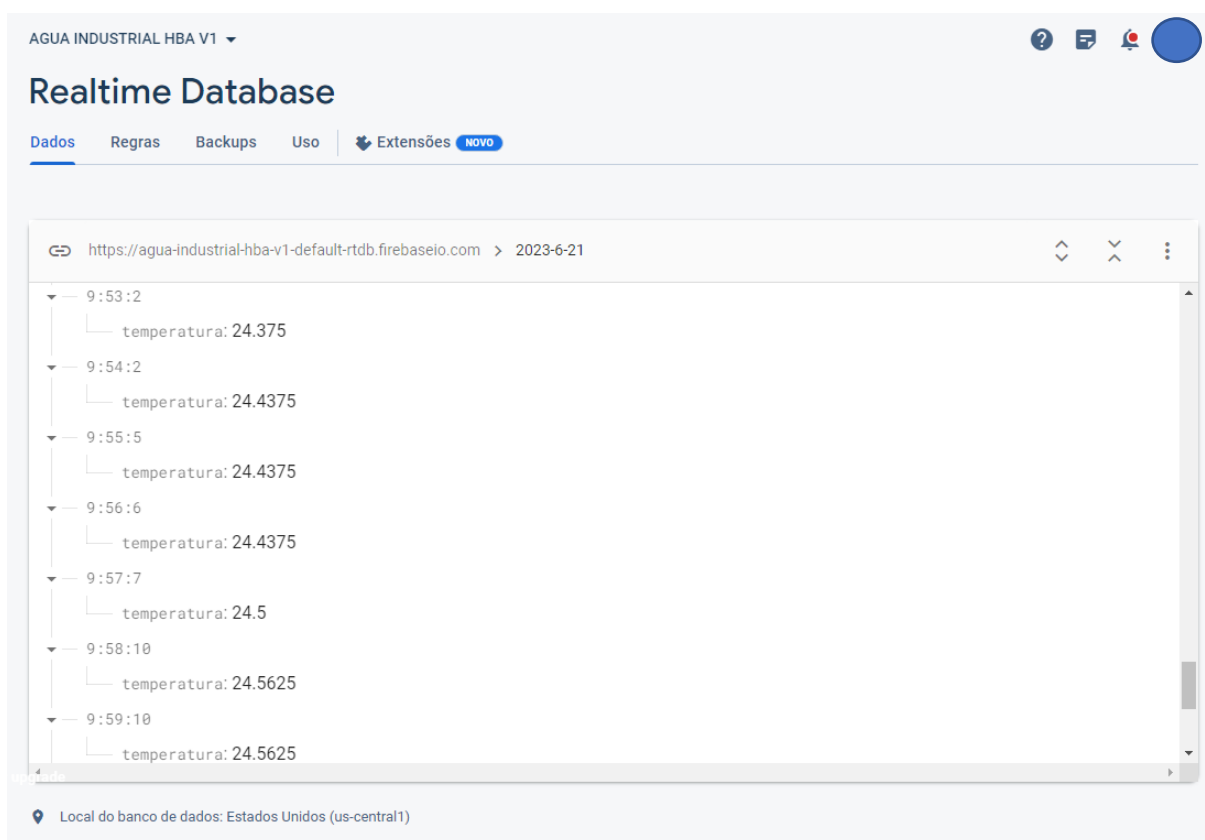
Figura 8- Teste inicial de bancada do sistema proposto



Fonte: Próprio autor.

Assim, utilizando o sistema criado e a programação realizada no microcontrolador ESP32, foi possível realizar a conexão via *WiFi* local e enviar os dados coletados para o banco de dados. A Figura 9 apresenta como foram dispostos no *Real Time Database*, os dados coletados no momento do teste inicial.

Figura 9- Dados de teste inicial enviados ao banco de dados



Fonte: Próprio autor.

É possível observar nessa captura de tela da Figura 9, que as amostras de teste vão de 9:53 até as 9:59, ou seja, com 1 minuto de intervalo em cada medição. É importante destacar que as medições foram realizadas em um maior período de tempo, porém analisou-se apenas a captura de tela.

É visível na Figura 9 que se criou um nó de data, sendo a data em questão 21 de junho de 2023, e posteriormente um nó que está inserido na data representando o horário local (Horário de Brasília, DF). Dentro de cada nó de horário, encontra-se um nó com a temperatura lida no horário em exatidão.

Observa-se que a temperatura da água utilizada nos testes no momento da leitura, era de 24,5°C em média.

4.5 IMPLIMENTAÇÃO NA INDÚSTRIA

Para implantar um sistema na indústria, é necessário levar em consideração alguns pontos que podem levar ao mal funcionamento do mesmo, até mesmo danificá-lo. Em uma rede industrial, se presencia problemas de interferências causados por harmônicas, ruídos e sobrecargas elétricas. Portanto é necessário ter alguns cuidados com o equipamento.

4.5.1 Local de implementação

A fim de testes em ambiente fabril, levou-se em consideração a geladeira industrial (*chiller*) presente na planta nº 2, onde é realizada extrusão de EPDM. A água é utilizada para resfriamento das unidades de forno micro-ondas industrial, assim como resfriamento das peças que passam por um processo de vulcanização do EPDM.

4.5.2 Materiais para a montagem industrial

Para o ambiente fabril, é necessário que o sistema esteja minimamente seguro, a fim de protegê-lo contra intemperes do ambiente físico, assim como dos surtos provenientes de uma rede elétrica industrial. Com o intuito de validação de equipamento e prototipagem, realizou-se uma montagem com alguns itens de reuso, como painel elétrico e fonte 24 Vdc. O desenho esquemático do painel montado é apresentado no Apêndice B.

Os materiais utilizados foram:

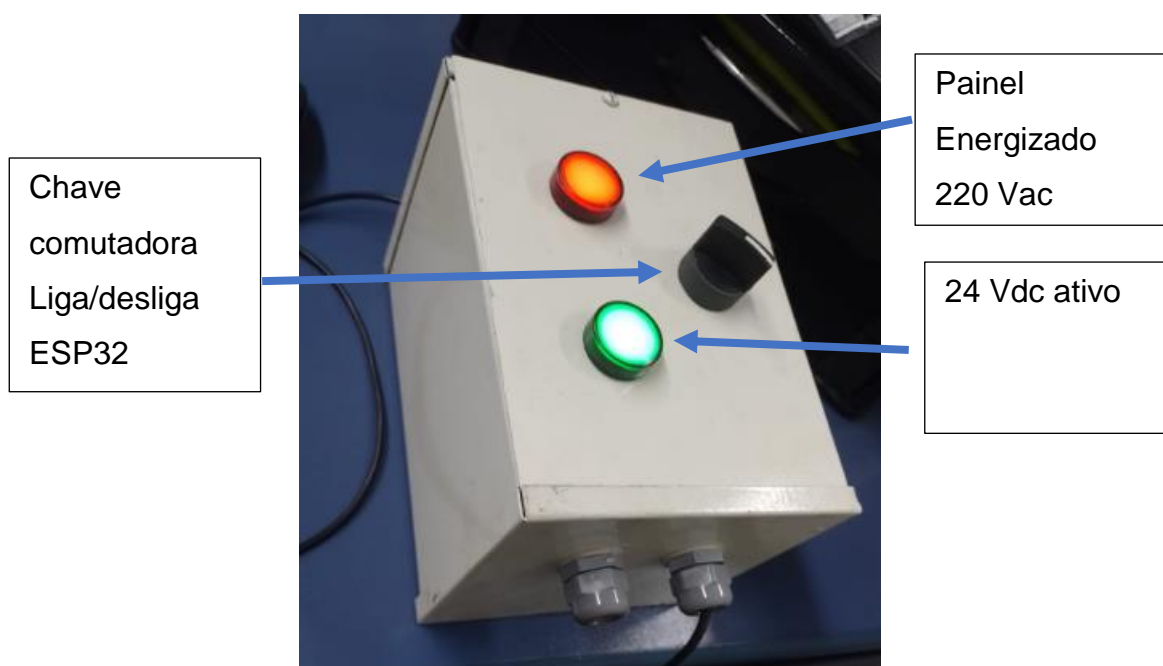
- Painel elétrico 300x150x150 mm
- Sinaleiro Luminoso 24 Vdc verde
- Sinaleiro Luminoso 220 Vac vermelho
- Chave comutadora de duas posições NA
- Disjuntor monopolar 2 A
- Fonte de alimentação Siemens LOGO! 24 Vdc 1,3 A
- Trilho DIM
- Placa microcontroladora ESP32

- Suporte de borne ESP32
- Regulador de tensão LM2596 com display StepDown

O painel elétrico é utilizado com intuito de realizar a prototipagem industrial, assim, pode-se validar o sistema levando em consideração as condições de uma fábrica. O painel utilizado foi de reuso, assim, poupamos recursos e reutilizamos materiais que seriam descartados.

Os sinaleiros são utilizados com o intuito de indicação de tensão e mostrar quando está definitivamente energizado. A Figura 10 apresenta a parte frontal do painel elétrico montado com os sinaleiros.

Figura 10- Externo do painel elétrico com o sistema de coleta de temperatura da água



Fonte: Próprio autor

No painel pode-se observar a presença do sinaleiro vermelho, que possui alimentação de 220 Vac e representa a energização do disjuntor geral. Já o sinaleiro verde, possui alimentação 24 Vdc, e representa a energização da fonte de alimentação.

A fonte de alimentação somente é energizada com a chave NA da chave comutadora, como é possível ver na parte frontal do painel elétrico.

É possível ver a parte interna do painel na Figura 11, baseado no Apêndice B.

Figura 11- Interno do painel elétrico com o sistema de coleta de temperatura da água



Fonte: Próprio autor

O circuito de regulador de tensão utilizado é o LM2596, que possui sua construção baseada nos conversores CC-CC do tipo Buck, com uma capacidade de condução de até 3 amperes e frequência de comutação de 150 kHz. A tensão de entrada do regulador é de 40 Vdc, o que torna compatível com a fonte de 24 Vdc de padrão industrial utilizada. Sua capacidade ajustável de tensão de saída, o torna perfeito para o projeto em questão, pois pode variar de 1,2 Vdc à 37 Vdc $\pm 4\%$, portanto, pode-se atingir a tensão necessária, que no caso do ESP32, é de 5 Vdc. Sua tensão de saída, pode ser regulada através de um trimpot (FONSECA, 2023).

4.5.3 Inicialização e conexão do dispositivo na rede industrial

Após a montagem, energizou todo o painel em etapas.

Inicialmente com o disjuntor desligado, verificou a tensão que energizará fonte dc, sendo 220 Vac entre fase e neutro em uma rede 380 Vac.

Assim, foi possível ligar o disjuntor e testar a tensão da fonte, em que sua saída será de 24 Vdc.

Com o microcontrolador desligado do painel, foi possível energizar o regulador de tensão StepDown, e na sequência, utilizando seu trimpot, ajustar sua saída para 5 Vdc.

Por fim, ligou-se o microcontrolador no conjunto.

4.5.4 Experimentação

Para realizar o a implementação do dispositivo, realizou-se algumas experimentações com o intuito de verificar a continuidade das medições, possíveis falhas ou divergências de temperatura. Para isso, foram considerados dois cenários. O primeiro cenário, foi deixar sensor de temperatura DS18B20 tipo sonda livre no ambiente, sem o contato com a água, feito no dia 24/10.

Considerou-se um padrão para realizar as medições, onde foi registrou-se 10 aquisições de temperatura em 10 minutos de experimento. Após a realização das medições, consultou-se o *Power Query* que faz o vínculo com o banco de dados. O Tabela 1 e a Figura 12 mostram o resultado da experiência no ambiente.

Tabela 1- Leituras de temperatura dia 24/10

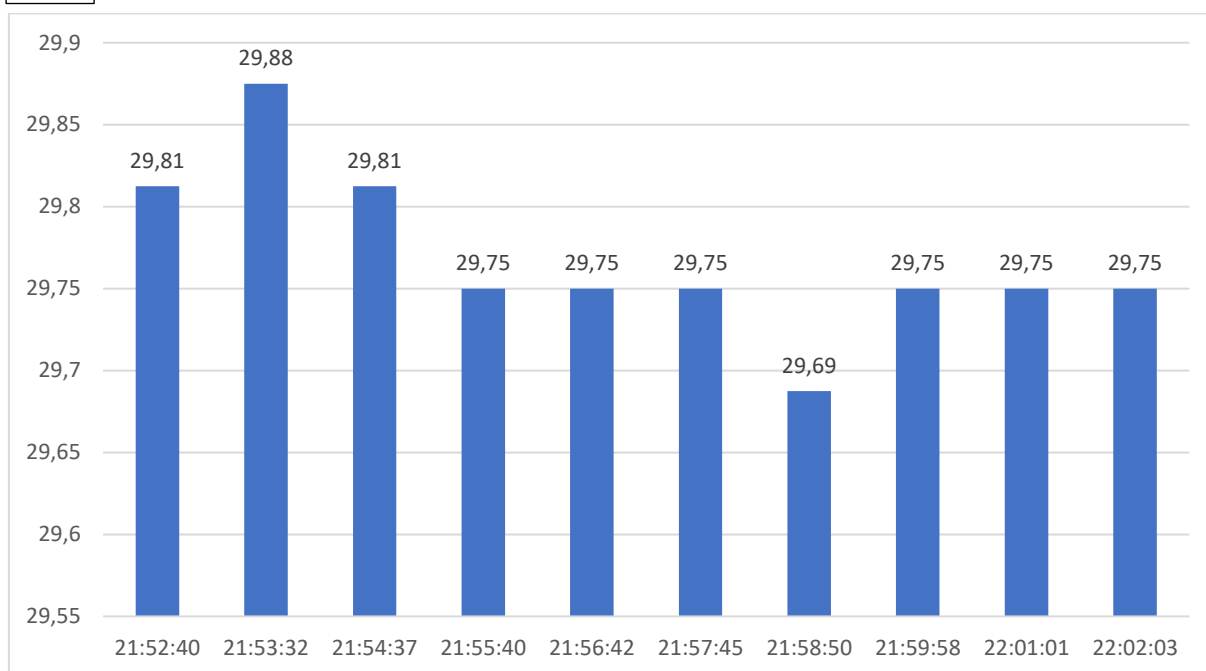
HORA	TEMPERATURA °C 24/10
21:52:40	29,81
21:53:32	29,87
21:54:37	29,81
21:55:40	29,75
21:56:42	29,75
21:57:45	29,75
21:58:50	29,69
21:59:58	29,75
22:01:01	29,75
22:02:03	29,75

Fonte: Próprio autor

Em sequência, tem-se o gráfico gerado a partir do quadro de temperaturas registradas.

°C

Figura 12- Gráfico de temperatura dia 24/10



Fonte: Próprio autor

Portanto, a média das medições ficaram em aproximadamente 29,80°C. As variações das temperaturas observadas dizem a respeito do erro que a sonda apresenta, assim como as mudanças pequenas de temperatura.

A fim de comparação, utilizou-se de outro medidor de temperatura, assim é possível analisar o erro percentual entre os dispositivos. Para isso, utilizou-se um medidor de temperatura “Fluke 62 MAX +”.

O medidor “Fluke 62 MAX +” tem a capacidade de trabalhar em uma faixa do ambiente operacional de 0°C à 50°C, com um erro de leitura de $\pm 1\%$ da temperatura ou ± 1 °C, o que for maior no momento da leitura, predomina. A Figura 13 mostra o momento de leitura enquanto a experiência estava sendo realizada.

Figura 13- Aquisição de temperatura dia 24/10 com medidor Fluke



Fonte: Próprio autor

Tem-se que a temperatura do medidor de temperatura é de 29,30°C.

Logo, calculando um erro percentual entre a temperatura da experiência e a temperatura do medidor Fluke, tem-se que:

$$E\%temp = \frac{|T_{Experimental} - T_{Fluke}|}{T_{Fluke}} * 100$$

$$E\%temp = \frac{|29,8 - 29,3|}{29,3} * 100 = 1,7\%$$

Considera-se 1,7% um erro percentual baixo para a aplicação em questão, já que a aferição não necessita em uma precisão decimal. Contudo, em aplicações mais precisas, seria necessário realizar um novo estudo, com testes de medição mais amplos, a fim de averiguar com precisão a leitura de temperatura do sensor tipo sonda.

O segundo cenário considerou-se inserir o sensor de temperatura DS18B20 tipo sonda em água gelada, assim, pode-se obter outra análise. A experiência foi realizada no dia 17/11.

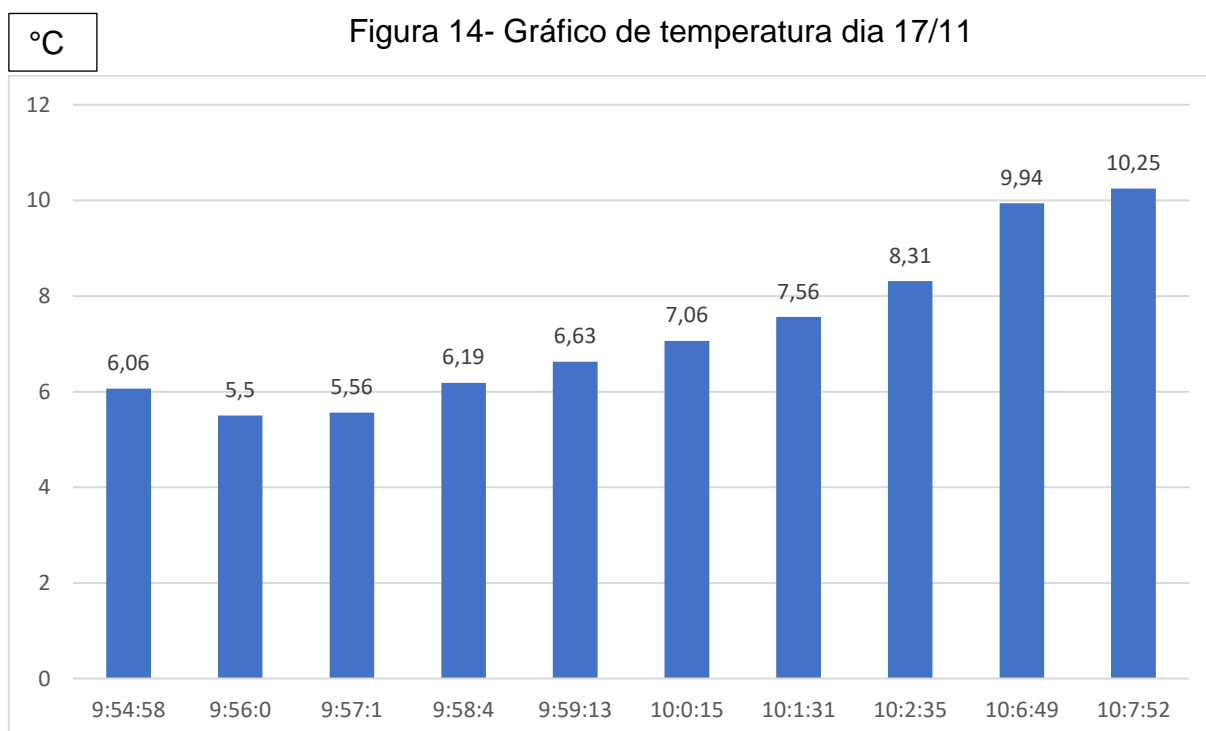
O Tabela 2 e a Figura 14 mostram o resultado da experiência em água gelada.

Tabela 2- Leituras de temperatura dia 17/11

HORA	TEMPERATURA °C 17/11
9:54:58	6,06
9:56:0	5,50
9:57:1	5,56
9:58:4	6,19
9:59:13	6,62
10:0:15	7,06
10:1:31	7,56
10:2:35	8,31
10:6:49	9,94
10:7:52	10,25

Fonte: Próprio autor

Em sequência, tem-se o gráfico gerado a partir do quadro de temperaturas registradas.



Fonte: Próprio autor

Portanto, a média das medições ficaram em aproximadamente 7,30°C. A crescente vista no gráfico da Figura 14, é o aumento da temperatura da água a medida em que a mesma troca calor com o ambiente. As variações das temperaturas observadas dizem a respeito do erro que a sonda apresenta, assim como as mudanças pequenas de temperatura.

Figura 15- Aquisição de temperatura dia 17/11 com medidor Fluke



Fonte: Próprio autor

Tem-se que a temperatura do medidor de temperatura é de 7,20°C.

Logo, calculando um erro percentual entre a temperatura da experiência e a temperatura do medidor Fluke, tem-se que:

$$E\%temp = \frac{|T_{Experimental} - T_{Fluke}|}{T_{Fluke}} * 100$$

$$E\%temp = \frac{|7,3 - 7,2|}{7,2} * 100 = 1,4\%$$

Considera-se 1,4% um erro percentual baixo para a aplicação em questão, já que a aferição não necessita em uma precisão decimal. Contudo, em aplicações mais

precisas, seria necessário realizar um novo estudo, com testes de medição mais amplos, a fim de averiguar com precisão a leitura de temperatura do sensor tipo sonda.

Após os resultados, foi possível ter a visão de como o sensor de temperatura DS18B20 tipo sonda se comporta em ambiente diferentes, assim como sua eficácia.

5. CONCLUSÃO

Este trabalho pretendeu entender o conceito de IoT na indústria, a fim de utilizar essa temática para elaborar um sistema que captasse a temperatura da água utilizada de indústria para processos de forma automática e online.

Para a aplicação e elaboração de um sistema capaz de realizar o processo, estudou-se a evolução da indústria e verificou-se toda trajetória tecnológica para que se chegasse à indústria 4.0. Nesse conceito, toda indústria deve possuir um funcionamento de forma automática, que está totalmente ligado ao conceito de telemetria, ou seja, a leitura real dos dados de forma online, sem esforços. Na aplicação estudada, a telemetria foi feita para leitura da temperatura da água industrial, com o intuito de analisar facilmente algo que já era realizado de forma manual, direcionando-se para a indústria 4.0.

Analizou-se o microcontrolador ESP32, utilizando-o no experimento. Pode-se confirmar sua eficiência na aplicação, pois foi favorável aos testes, além de ser um equipamento de baixo custo. Verificou-se que a utilização do banco de dados *Firebase Real Time Database* foi conveniente ao projeto, pois é de fácil acesso e uso, além de ser gratuito, todavia, o formato de arquivos utilizado por ele (JSON), não é abrangente às aplicações atuais, o que dificulta o processo.

Em relação à aplicação industrial, pode-se destacar o ambiente como degradante ao equipamento, portanto é necessário a implementação de equipamentos mais robustos e que tenham um grau de proteção maior contra intempéries do meio industrial, assim como é necessário a proteção à rede elétrica industrial, a fim do equipamento ter uma longevidade.

Observou-se com as experimentações, que o equipamento trabalha de forma adequada e atende ao necessário, já que esse tipo de aplicação não envolve uma precisão decimal.

Para o melhor tratamento e utilização dos dados futuros, pode-se direcionar as leituras de temperatura para um banco de dados mais prático, e conseqüentemente, pode-se implementar um melhor sistema de monitoramento, podendo evoluir para um sistema supervisorio completo da planta industrial. Assim como também se pode criar em pontos estratégicos, telas remotas para leitura dos dados em tempo real.

Outro ponto de melhoria que vale a pena destacar, é melhorar a proteção do equipamento da rede elétrica industrial, tentando isolar das redes de utilização dos

motores e equipamentos de atrapalham um bom funcionamento e longevidade de eletrônicos na indústria.

REFERÊNCIAS

BABIUCH, Marek; FOLTÝNEK, Petr; SMUTNÝ, Pavel. **Using the ESP32 Microcontroller for Data Processing**. Krakow-Wieliczka, Poland: IEEE, 2019. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8765944>. Acesso em: 8 de junho de 2023.

DOCS. **Using the Arduino Software (IDE) | Arduino Documentation**. Disponível em: <https://docs.arduino.cc/learn/starting-guide/the-arduino-software-ide>>. Acesso em: 27 de maio de 2023.

ELETROGATE. **Guia Completo do Sensor DS18B20 a Prova D'água**. Universidade estadual de Ponta Grossa- UEPG (2019). Disponível em: <https://blog.eletrogate.com/guia-completo-sobre-sensor-de-temperatura-ds18b20-a-prova-dagua/>>. Acesso em: 27 de maio de 2023.

ESP-IDF. **ESP-IDF Programming Guide: latest documentation**. Disponível em: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>>. Acesso em: 27 de maio de 2023.

ESPRESSIF. **ESP32 Series DataSheet Including**. [s.l: s.n.]. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_DataSheet_en.pdf>. Acesso em: 27 de maio de 2023.

FEREIRA, I, V.; BIGHETI, J, A.; MANSANO, R, K.; GODOY, E, P. **PROPOSTA DE UM MODELO PARA A APLICAÇÃO DA INTERNET DAS COISAS INDUSTRIAL**. XIII Simpósio Brasileiro de Automação Inteligente, Porto Alegre, outubro, 2017.

FIREBASE. **Get Started with Firebase Realtime Database for C++**. Disponível em: <https://firebase.google.com/docs/database/cpp/start?hl=pt-br>>. Acesso em: 6 jun. 2023.

FONSECA, R, M, S. **TELEMETRIA E INOVAÇÃO EM SISTEMAS DE INFORMAÇÃO NA GESTÃO DE ESTOQUES: estudo do caso da indústria Vale** **LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator**. [s.l: s.n.]. Disponível em: <https://www.ti.com/lit/ds/symlink/lm2596.pdf>>.

LOPEZ RESEARCH LLC. **“Uma introdução à Internet da Coisas (IoT)”**. Disponível em: https://www.cisco.com/c/dam/global/pt_br/assets/brand/iot/iot/pdfs/lopez_research_an_introduction_to_iiot_102413_final_portuguese.pdf>. Acesso em: 4 dez. 2023.

MAXIM INTEGRATED. **DS18B20 DataSheet, 2015**. Disponível em: <https://DataSheets.maximintegrated.com/en/ds/DS18B20.pdf>. Acesso em: 27 de maio de 2023.

NEOWATER. **Sistema de telemetria via IoT pode combater o desperdício e ajudar empresas a economizar água.** Campinas, 2021. Disponível em: <https://www.neowater.com.br/post/telemetria-iot>. Acesso em: 7 de março de 2023.

SAKURAI, R.; ZUCHI, J. D. **REVOLUÇÕES INDUSTRIAIS ATÉ A INDÚSTRIA 4.0.** Revista Interface Tecnológica, v. 15, n. 2, p. 480–491, 2018.

STANFORD, HERBERT, W. **HVAC Water *Chillers and Cooling Towers*:** Fundamentals, Application and Operation. 1ª Ed New York: Marcel Dekker, 2003. 309 p.

APÊNDICE A – Programação desenvolvida no Arduino IDE para o ESP32

```

#include <WiFi.h>
#include <FirebaseESP32.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <TimeLib.h>

// Define as credenciais da rede Wi-Fi
const char* ssid = "celularg20";
const char* password = "12345678";

// Define as credenciais do projeto Firebase
#define FIREBASE_HOST "*****"
#define FIREBASE_AUTH "*****"

// Configuração do sensor DS18B20
#define ONE_WIRE_BUS 17
OneWire OneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20(&OneWire);
float tempC;

// Configuração do cliente NTP para obtenção da hora
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");

// Fuso horário de Brasília (UTC-3 durante o horário padrão e UTC-2 durante o
horário de verão)
int timeZoneOffset = -3 * 3600; // Offset em segundos

// Crie uma instância da classe FirebaseData

FirebaseData fbdo;
void setup() {
  Serial.begin(115200);

  // Conecta-se à rede Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.print("Conectado! Endereço IP: ");
  Serial.println(WiFi.localIP());

```

```

// Inicializa a biblioteca Firebase
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

// Inicializa o sensor DS18B20
DS18B20.begin();

// Inicializa o cliente NTP para obtenção da hora
timeClient.begin();
timeClient.setTimeOffset(timeZoneOffset);

// Sincroniza a hora usando o cliente NTP e a biblioteca TimeLib
setSyncProvider(getNtpTime);
setSyncInterval(300); // Sincroniza a cada 5 minutos
}

void loop() {
  // Coletar o valor da temperatura do sensor DS18B20
  DS18B20.requestTemperatures();
  float temperatura = DS18B20.getTempCByIndex(0);

  // Obter a data e hora atual no fuso horário de Brasília
  time_t t = now();
  String timestamp = String(hour(t)) + ":" + String(minute(t)) + ":" +
String(second(t));
  String data = String(year(t)) + "-" + String(month(t)) + "-" + String(day(t));

  // Criar um novo nó com o horário atual
  String nodePath = "/" + data + "/" + timestamp;

  // Definir o valor da temperatura dentro do novo nó
  Firebase.setFloat(fbdo, nodePath + "/Temperatura", temperatura);

  Serial.print("Temperatura: ");
  Serial.print(temperatura);
  Serial.print("°C");
  Serial.println();

  delay(1800000); // Esperar 30 minutos antes de enviar os próximos dados
}

time_t getNtpTime() {
  timeClient.update();
  return timeClient.getEpochTime();
}
}

```

APÊNDICE B – Diagrama elétrico para instalação no painel industrial feito do SEE Electrical

