



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

LEONARDO UEMURA

**SISTEMA EMBARCADO PARA A COLETA DE DADOS DE ESTAÇÃO
METEOROLÓGICA PARA CAMPUS INTELIGENTE**

Sorocaba

2023

LEONARDO UEMURA

SISTEMA EMBARCADO PARA A COLETA DE DADOS DE ESTAÇÃO
METEOROLÓGICA PARA CAMPUS INTELIGENTE

Trabalho de Conclusão de Curso de
Graduação apresentado como parte dos
pré-requisitos para a obtenção do título de
Engenheiro de Controle e Automação, à
Universidade Estadual Paulista, “Júlio de
Mesquita Filho”.

Orientador:

Prof. Dr. Eduardo Paciência Godoy

Sorocaba

2023

U22s Uemura, Leonardo
Sistema embarcado para a coleta de dados de estação meteorológica para campus inteligente / Leonardo Uemura. -- Sorocaba, 2023
69 p.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Eduardo Paciência Godoy

1. Aquisição de dados. 2. Internet das coisas. 3. Cidades inteligentes. 4. Automação Industrial - Protocolos. 5. Estações meteorológicas. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.



**UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"**

Instituto de Ciência e Tecnologia de Sorocaba

**SISTEMA EMBARCADO PARA A COLETA DE DADOS DE ESTAÇÃO
METEOROLÓGICA PARA CAMPUS INTELIGENTE**

LEONARDO UEMURA

**ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO
PARTE DO REQUISITO PARA A OBTENÇÃO DO GRAU DE
BACHAREL EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

Prof. Dr. Everson Martins
Coordenador

BANCA EXAMINADORA:

Prof. Dr. EDUARDO PACIENCIA GODOY
Orientador / UNESP – Campus Sorocaba

Prof. Dr. Luis Armando de Oro Arenas
Avaliador / UNESP – Campus Sorocaba

Prof. Dr. Antonio Cesar Germano Martins
Avaliador / Externo

Sorocaba

2023

dedicado ao meu pai Eduardo,
que me ensinou honestidade,
paciência e razão.

AGRADECIMENTOS

Agradeço minha mãe Rosana e meu pai Eduardo, meus principais apoiadores, por terem me proporcionado a oportunidade de cursar Engenharia de Controle e Automação, sempre me fornecendo ensinamentos e incentivos nos períodos que antecederam minha escolha de graduação de forma a proporcionar o conhecimento necessário para o acompanhamento devido do curso. Agradeço à minha namorada, Patricia, que sempre esteve presente me incentivando a de forma compreensiva e sempre me auxiliando durante o desenvolvimento da minha pesquisa. Aos meus companheiros que moraram comigo, durante todo o curso, Julio, Seiry, Judas, Guita e Mandela, por estarem sempre presentes na minha vida me auxiliando a passar por todos os obstáculos durante a trajetória universitária, sempre me proporcionando ótimas experiências. Minhas conquistas e minha história se devem a vocês. Agradeço imensamente ao professor Eduardo Paciência, que ministrou as matérias que hoje servem como base para a minha carreira fora da área acadêmica, despertando o meu interesse pela área, e por todo o auxílio ao longo do projeto, me incentivando a não desistir e me auxiliando a encontrar formas de realizar esse trabalho com muita dedicação e sempre se encontrando disponível, mesmo fora do horário comercial. Isso foi essencial para a conquista de um resultado mais refinado nesse projeto. Agradeço também aos meus mentores, Raphael Costa, Diego Mariano e Rafael Mendes, que me forneceram conhecimentos profundos na área de engenharia e programação. Minha inspiração se baseia em vocês.

UEMURA, L. **Sistema Embarcado para a coleta de dados de Estação Meteorológica para Campus Inteligente**. 2023. Trabalho de Graduação (Engenharia de Controle e Automação) — Instituto de Ciência e Tecnologia de Sorocaba, UNESP - Universidade Estadual Paulista, Sorocaba, 2023.

RESUMO

A integração de dados entre diferentes sistemas é a etapa principal para a realização da coleta e historiamiento das informações de dispositivos IoT em arquiteturas inteligentes, fornecendo uma ferramenta de tomada de decisões baseada em evidências por meio do monitoramento dos dados em tempo real de um equipamento. Com o intuito de se padronizar o modo com o qual estes dispositivos compartilham informações, foram implementadas regras de comunicação, que são os protocolos de comunicação industriais. A Internet das Coisas refere-se a uma rede de dispositivos físicos que utilizam a conectividade realizada por protocolos de comunicação industrial. Aproveitando-se da infraestrutura IoT do Smart Campus da Unesp Sorocaba, este trabalho implementou uma nova integração de dados entre a arquitetura inteligente e a estação meteorológica existente no campus, utilizando um sistema embarcado para realizar a ponte entre dispositivos e sistemas se aproveitando de diferentes protocolos de comunicação industriais, com o intuito da digitalização seus dados climáticos. Este dispositivo realiza o recebimento dos dados da estação meteorológica utilizando a serial do datalogger CR800 Series da empresa Campbell Scientific e enviando-o via MQTT à arquitetura IoT do campus inteligente. Em complemento à criação do sistema embarcado, foram desenvolvidas telas interativas dentro da página web do campus inteligente que podem ser acessadas pela comunidade e configurado um sistema de alertas para a notificação de possíveis problemas operacionais da estação meteorológica, como alertas de mal funcionamento. A implementação realizada digitalizou a estação meteorológica, fornecendo a visualização em tempo real dos dados históricos climáticos, facilitando o monitoramento das informações pela comunidade acadêmica e realização de análises de correlação de variáveis climáticas.

Palavras-chave: integração de dados; internet das coisas; smart campus; estação meteorológica; serial; CR800 Series; datalogger; mqtt.

UEMURA, L. **Embedded System for Meteorological Station Data Collection for Smart Campus**. 2023. Bachelor Work (Control and Automation Engineering) — Institute of Science and Technology of Sorocaba, UNESP - Universidade Estadual Paulista, Sorocaba, 2023.

ABSTRACT

The data integration between different systems is the main step for collecting and storing information from IoT devices in intelligent architectures, providing an evidence-based decision-making tool through real-time data monitoring of a device. In order to standardize the way these devices share information, communication rules have been implemented, known as industrial communication protocols. The Internet of Things refers to a network of physical devices that utilize connectivity provided by industrial communication protocols. Taking advantage of the robust IoT infrastructure of Unesp Sorocaba's Smart Campus, this work implemented a new data integration between the intelligent architecture and the existing weather station on campus, using an embedded system to bridge devices and systems using different industrial communication protocols, with the aim of digitizing its weather data. This embedded system receives data from the weather station using the serial port of the Campbell Scientific CR800 Series datalogger and sends it via MQTT to the IoT architecture of the smart campus. In addition to the development of the embedded system, interactive screens were created within the smart campus web page, which can be accessed by the community, and an alert system was configured to notify possible operational issues with the weather station, such as malfunction alerts. The implemented solution digitized the weather station, providing real-time visualization of historical weather data, facilitating information monitoring by the academic community and enabling analysis of correlations between weather variables.

Keywords: data integration; internet of things; smart campus; weather station; serial; CR800 Series; datalogger; mqtt.

LISTA DE FIGURAS

Figura 1 - Celular conectado aos dispositivos por meio da rede.	16
Figura 2 - Fluxograma do funcionamento do MQTT	20
Figura 3- Estação Meteorológica e seus sensores.....	22
Figura 4 - Estação Compacta	23
Figura 5 - Pluviômetro	24
Figura 6 - Piranômetro	25
Figura 7 - Topologia da aplicação	28
Figura 8 - Painel elétrico do Smart Campus com régua de alimentação fixada na parede.	29
Figura 9 - Estação após organização dos fios.	29
Figura 10 - Alimentação da estação meteorológica com emenda de derivação para alimentação do microcontrolador.	30
Figura 11 - Porta RS232 e porta COM1 com utilização dos terminais TX e GND.....	31
Figura 12 - Configuração do datalogger CR800 no software PC400.	32
Figura 13 - Ferramentas nativas do PC400.....	32
Figura 14 - Programação do datalogger utilizando o CRBasic Editor.....	33
Figura 15 - Fluxograma do funcionamento do código do datalogger.....	34
Figura 16 - Adição da lógica de envio de dados pela serial da porta COM1.	35
Figura 17 - Configurações avançadas utilizando o Device Configuration Utility.....	36
Figura 18 - Fluxograma do código carregado no microcontrolador.	38
Figura 19 - Formato do dado já interpretado enviado pela serial.	39
Figura 20 - Diagrama de conexão serial entre o datalogger e o microcontrolador.....	40
Figura 21 - Exemplo de estrutura JSON esperada pelo Smart Campus.	41
Figura 22 - Invólucro feito por impressão 3D.	42
Figura 23 - Vista interna do invólucro com a fixação do protoboard e conexões dos fios.	43
Figura 24 - Consulta com agregação de acordo com o intervalo selecionado na tela.	47
Figura 25 - Consulta com agregação de soma de um período de um dia.	47
Figura 26 - Consulta com retorno de apenas o último resultado.	48
Figura 27 - Gráfico customizado da velocidade e direção do vento.....	48
Figura 28 - Consulta com agregação de duas variáveis e tabela retornada com as duas variáveis no mesmo timestamp.....	49

Figura 29 - Código Javascript responsável pela renderização customizada do painel.	50
Figura 30 - Escolha do nome e plataforma gerenciadora de alertas.	51
Figura 31 - Lógica de ativação do alerta.	52
Figura 32 - Intervalo de verificação e detalhes do alerta.	52
Figura 33 - Saúde do alerta cadastrado.	53
Figura 34 - Sistema embarcado instalado, apresentado pela caixa azul.	55
Figura 35 - Parte superior da Tela 1.	56
Figura 36 - Descrição das variáveis climáticas.	56
Figura 37 - Parte Inferior da Tela 1.	57
Figura 38 - Dados históricos de temperatura.	58
Figura 39 - Dados históricos de radiação solar.	59
Figura 40 - Dados históricos de direção e velocidade do vento.	59
Figura 41 - Dados históricos de Umidade.	60
Figura 42 - Dados históricos de Pressão e Precipitação.	60
Figura 43 - Responsividade mobile para temas claro e escuro.	61

LISTA DE QUADROS

Quadro 1 - Características do Campus Digital e do Campus Inteligente.....	22
---	----

LISTA DE TABELAS

Tabela 1- Equipamentos da Estação Meteorológica.....	23
Tabela 2 - Detalhamento das medidas físicas dos sensores da Estação Meteorológica	27

LISTA DE ABREVIATURAS E SIGLAS

IoT - Internet of Things
MQTT - MQ Telemetry Transport
IP - Internet Protocol
UNESP - Universidade Estadual Paulista
ICTS - Instituto de Ciência e Tecnologia - Câmpus de Sorocaba
Wi-Fi - Wireless Fidelity
RAM- Random Access Memory
LAN- Local Area Network
JSON - JavaScript Object Notation
HTML - HyperText Markup Language
CSS - Cascading Style Sheet

SUMÁRIO

1	INTRODUÇÃO E JUSTIFICATIVA	13
2	REFERENCIAL TEÓRICO	16
2.1	Internet das Coisas (IoT)	16
2.2	Smart Campus	17
2.3	Sistemas Embarcados	18
2.4	Esp32 Wroom 32U	19
2.5	Protocolo MQTT	19
3	SISTEMA DA ESTAÇÃO METEOROLÓGICA	21
3.1	Localização e características gerais	21
3.2	Estação Compacta	23
3.3	Pluviômetro	24
3.4	Piranômetro	24
3.5	Comunicação dos dados	25
3.6	Informações transmitidas	26
4	DESENVOLVIMENTO DO SISTEMA EMBARCADO	28
4.1	Abordagem	28
4.2	Manutenção e reprogramação da Estação Meteorológica	28
4.3	Sistema Embarcado	36
4.3.1	Fluxo do funcionamento adotado para o código	37
4.3.2	Estrutura de recebimento das variáveis pela serial	39
4.3.3	Diagrama de conexão da estação	39
4.3.4	Código .ino do esp32	40
4.3.4	Testes de comunicação	41
4.4	Montagem do Invólucro	42
4.4.1	Requisitos do invólucro	42
4.4.2	Conexão dos fios	43
4.5	Interface Web de Visualização	43
4.5.1	Princípios básicos para as visualizações dos dados	45
4.5.2	Casos especiais obtidos	46
4.6	Configuração de aviso caso o sistema pare de enviar dados	50

5	RESULTADOS	54
5.1	Sistema Embarcado	54
5.2	Painel e interações do usuário.....	55
5.2.1	Tela 1 – Informações instantâneas	55
5.2.1	Tela 2 – Informações históricas	57
6	CONCLUSÕES	62
	REFERÊNCIAS	64

1 INTRODUÇÃO E JUSTIFICATIVA

A pandemia de Covid-19, declarada em março de 2020 pela Organização Mundial da Saúde, intensificou o uso de tecnologias digitais no país, passando de 71% dos domicílios com acesso a internet para 83% de 2019 a 2020, correspondendo a um aumento de 61.8 milhões de domicílios com acesso a rede (NITAHARA, 2021). Com esse crescimento de pessoas conectadas à rede, também se evoluiu a disseminação do conceito e uso da Internet das Coisas.

A Internet das Coisas é uma rede de objetos físicos capazes de se conectarem e transmitir dados por meio de uma comunicação sem fio. Em outras palavras, o aparelho é capaz de transmitir dados para uma solução digital, da mesma forma que acontece entre dispositivos conectados à internet (TOTVS, 2022). Com o uso de novas tecnologias para o transporte de informações, gerou-se a necessidade de um maior controle sobre a forma com a qual os dispositivos devem se comunicar, visando uma padronização no formato da comunicação para que dispositivos de diferentes marcas e modelos possam trocar informações facilmente.

Um protocolo define a estrutura de mensagens que devem ser reconhecidas e utilizadas pelos controladores, independentemente do tipo de rede em que se comunicam. Ele descreve o processo que um controlador usa para solicitar acesso a outro dispositivo, como ele deve responder às solicitações de acesso a informações e a forma com a qual os erros devem ser detectados e reportados, estabelecendo um formato padronizado para o arranjo e conteúdo dos campos da mensagem.

A integração de diferentes sistemas muitas vezes exige a utilização de variados protocolos industriais, de forma que múltiplos equipamentos de diversas marcas e modelos possam operar em conjunto sem perda de informação. Diversos entraves de implementação de projetos de Internet das Coisas se encontram nesse contexto, no qual o cliente precisa se atualizar ou adequar a comunicação de seus equipamentos para alcançar processos que envolvem o monitoramento de dados e possivelmente sejam tomadas ações para melhoria de processos (ESBRASIL, 2023). É nesse contexto desafiador que a integração da estação meteorológica e muitos outros projetos de IoT se encontram.

Com o atual desenvolvimento da ciência e tecnologia, a tecnologia de big data, Internet das coisas, a computação em nuvem e a tecnologia da Internet móvel fizeram grandes progressos. Nesse contexto, a construção do Smart Campus tem gradualmente atraído a atenção das pessoas, especialmente nas universidades públicas, assumindo a liderança na construção de candidaturas. O atual Smart Campus é a atualização do campus digital, o que pode

proporcionar uma vida melhor e ambiente acadêmico para os professores universitários e estudantes. (XIONG, 2016)A Unesp Sorocaba, inaugurou em 2022 seu próprio Smart Campus e já possui projetos de monitoramento de dados como a integração dos painéis fotovoltaicos instalados no prédio novo, captando de forma flexível, escalável e segura as informações coletadas pelo inversor do sistema fotovoltaico (MORAES, 2022).

Outras faculdades públicas também participaram do movimento de criação de Smart Campus como a UNICAMP, que implantou um projeto chamado Smart Lock, em que foi desenvolvido uma fechadura eletrônica que controla o acesso de pessoas utilizando cartões de rádio frequência (RFID) e QR Code (CLETO, 2021).

1.1 Objetivos

O presente trabalho tem como objetivo o desenvolvimento de um sistema embarcado para a coleta e integração de dados de uma estação meteorológica na arquitetura do campus inteligente da UNESP Sorocaba e visualização das informações climáticas coletadas em interfaces web interativas, facilitando a consulta ao sistema pela comunidade acadêmica.

1.2 Organização deste trabalho

O trabalho foi organizado em 7 capítulos, com o intuito de apresentar um estudo completo sobre a implementação de uma integração de dados da estação meteorológica ao sistema Smart Campus.

O primeiro capítulo, "Introdução e objetivos", apresenta a justificativa para a realização do trabalho, os objetivos que serão alcançados e a organização adotada para desenvolvê-los.

O segundo capítulo, "Referencial teórico", aborda os conceitos fundamentais que embasam a integração da estação meteorológica, tais como conceitos de internet das coisas, campus inteligente, sistema embarcado, técnicas de comunicação de dados, além de uma revisão dos trabalhos relacionados que serviram de base para o desenvolvimento do projeto.

O terceiro capítulo, "Sistema da Estação Meteorológica", apresenta os componentes utilizados na construção da estação meteorológica, como os sensores e placas de aquisição de dados, além de detalhar as características do sistema de comunicação de dados e dos softwares necessários para o funcionamento do sistema.

O quarto capítulo, "Desenvolvimento", descreve a arquitetura proposta para a integração dos dados coletados pelos sensores da estação meteorológica no sistema embarcado, detalhando as soluções adotadas para a aquisição, tratamento e armazenamento dos dados.

Também é demonstrado o processo de montagem do invólucro para proteção do sistema embarcado e o desenvolvimento dos painéis que compõem os dashboards interativos que exibem as informações meteorológicas em tempo real, possibilitando o monitoramento remoto do sistema.

Os capítulos cinco e seis, "Resultados e Conclusão", trazem os resultados obtidos na implementação do sistema, apresentando gráficos e tabelas para análise e discussão dos dados coletados.

Ao final, o trabalho apresenta as "Referências bibliográficas e anexos", que incluem a lista de referências utilizadas na elaboração do trabalho, bem como os anexos contendo detalhes técnicos sobre a arquitetura da estação meteorológica e do sistema embarcado.

2 REFERENCIAL TEÓRICO

Este capítulo contém o embasamento teórico utilizado no trabalho, com uma breve revisão dos conceitos básicos de Internet das Coisas (IoT) para uma melhor compreensão da concepção de um Smart Campus. Também serão esclarecidas as teorias necessárias ao desenvolvimento de um sistema embarcado, do dispositivo utilizado e sua comunicação com a plataforma inteligente.

2.1 Internet das Coisas (IoT)

A Internet das Coisas (IoT) é uma tecnologia que permite que dispositivos físicos se conectem e se comuniquem entre si por meio de uma rede. Com mais de 7 bilhões de dispositivos IoT conectados hoje, os especialistas esperam que esse número cresça para 10 bilhões em 2020 e 22 bilhões em 2025 (ORACLE, 2019). É uma tendência tecnológica em rápido crescimento que tem o potencial de revolucionar a maneira como vivemos, trabalhamos e interagimos com nosso ambiente. O conceito de IoT é baseado na ideia de conectar objetos do cotidiano à internet, permitindo que eles se comuniquem entre si e com os humanos. Essa tecnologia tem muitas aplicações em potencial, desde casas e cidades inteligentes até saúde e transporte. Os dispositivos IoT são incorporados com sensores, software e conectividade de rede, permitindo que eles coletem e troquem dados. A Figura 1 a seguir exemplifica a conectividade na palma da mão.

Figura 1 - Celular conectado aos dispositivos por meio da rede.



Fonte: MENDES, 2020

Os benefícios da IoT são numerosos e variados e dependem da aplicação específica. Alguns dos benefícios mais significativos incluem eficiência, proteção e segurança, economia de custos e tomada de decisão aprimoradas. No entanto, a IoT também enfrenta desafios como segurança, privacidade e interoperabilidade dependendo da sua implementação.

Uma das aplicações mais promissoras da IoT é o conceito de campus inteligente. O qual pode ser considerado como um ecossistema conectado e inteligente que usa a IoT para otimizar as operações, aumentar a segurança e melhorar a qualidade de vida de alunos, professores e funcionários (Bandeira, 2020). Ao usar dispositivos IoT para monitorar e analisar dados, um campus inteligente pode fornecer informações valiosas que podem ser usadas para tomar melhores decisões.

2.2 Smart Campus

O conceito de Smart Campus ou "Campus Inteligente" ganhou atenção crescente nos últimos anos, à medida que as universidades e outras instituições de ensino superior buscam melhorar sua eficiência, sustentabilidade e experiência geral no campus. Um Campus Inteligente utiliza tecnologias inovadoras, como a Internet das Coisas (IoT), inteligência artificial e análise de big data para conectar e otimizar recursos e operações do campus.

A estrutura de aplicação de um Smart Campus é uma combinação de IOT e computação em nuvem com base na alta computação de desempenho e internet. Por meio da integração dos sistemas como uma plataforma unificada, com exemplos de gestão de sistema de ensino, sistema financeiro, sistema de escritório, gerenciamento de sistema de biblioteca e gerenciamento de equipamentos, o campus inteligente pode fazer professores, alunos, pais, empresas e pesquisadores estudarem e se comunicarem. (NIE, 2013)

Salienta-se que um campus inteligente (Smart Campus) se difere do conceito de campus digital (Digital Campus). Destaca-se pontos relevantes destas distinções no quadro 1 a seguir. (PROENÇA, 2022). O Campus Digital enquadra-se em um estágio prévio ao Smart Campus, em que a conexão é local, sem o uso de IoT e utilizando o armazenamento local, fatores que resultam em um sistema de gestão isolado.

Quadro 1 – Características do Digital Campus e do Smart Campus

	Campus Digital	Smart Campus
Ambiente Técnico	<ul style="list-style-type: none"> • Rede Local • Internet 	<ul style="list-style-type: none"> • IoT • Armazenamento em nuvem • Rede sem fio • Internet Móvel
Aplicação	<ul style="list-style-type: none"> • Biblioteca digital • Educação a Distância • Ensino digital 	<ul style="list-style-type: none"> • Biblioteca inteligente • Ensino com realidade aumentada • <i>Smart Classroom</i>
Sistema de gestão	<ul style="list-style-type: none"> • Sistema isolado 	<ul style="list-style-type: none"> • Sistema de compartilhamento • Sistema Inteligente

Fonte: PROENÇA, 2020

Por meio da integração de sensores inteligentes, análise de dados e sistemas de automação, um Smart Campus pode oferecer uma série de benefícios para a comunidade ao redor. Esses benefícios podem incluir consumo de energia otimizado, segurança e proteção aprimoradas, sistemas de transporte refinado e experiências de aprendizado personalizadas.

A implementação de um Smart Campus requer uma estratégia abrangente e colaboração entre várias partes interessadas, incluindo departamentos de TI, gerenciamento de instalações e departamentos acadêmicos. Também envolve abordar vários desafios, como privacidade e segurança de dados, interoperabilidade de sistemas e equilibrar as necessidades de diferentes constituintes do campus.

O desenvolvimento de um Smart Campus representa uma oportunidade significativa para as instituições de ensino superior criarem ambientes mais sustentáveis, eficientes e inovadores que apoiem o crescimento acadêmico e pessoal dos membros de suas comunidades.

2.3 Sistemas Embarcados

Um sistema embarcado é um tipo de sistema computacional que consiste em hardware e software, projetado para executar funções específicas em um sistema maior. Esses sistemas são integrados em outros produtos ou equipamentos com o objetivo de controlar ou monitorar processos ou funções específicas. Geralmente, são desenvolvidos para serem simples e de baixo custo, com um conjunto limitado de componentes, como microcontroladores, sensores e atuadores. Ademais, os sistemas embarcados são projetados para serem compactos, eficientes em termos de energia e confiáveis. Eles precisam lidar com restrições de espaço e recursos,

como memória e processamento limitados, e podem ser programados para tomar decisões em resposta a entradas de sensores ou outras condições (SOUZA, 2023).

Os sistemas embarcados podem ser encontrados em diversas aplicações, como sistemas de segurança, dispositivos médicos, automóveis, equipamentos eletrônicos, , entre outros desenvolvidos para executar funções específicas de maneira automatizada e precisa.

Esses sistemas atuam um papel crucial em diversos aspectos da vida cotidiana, desde sistemas de infraestrutura até a vida pessoal. Com a contínua evolução da tecnologia, esses sistemas desempenham um papel ainda mais importante na sociedade moderna, se tornando cada vez mais sofisticados e capazes.

2.4 Esp32 Wroom 32U

O ESP32 WROOM 32U é um microcontrolador da família esp, que possui WiFi, Bluetooth e possui o diferencial de conexão opcional a uma antena externa para a melhoria da recepção de sinal. Esse modelo possui um cristal de 40 MHz e placa de dimensões de 18 x 19.2 x 3.2 mm. Seu CPU é um dual-core Xtensa 32-bit LX6 MCU, permitindo o controle individual de cada core por meio da programação carregada ao microprocessador. Esse chip suporta condições térmicas de operação de -40°C a 85°C e tensão de alimentação de 3 a 3,6V (ESPRESSIF, 2023).

2.5 Protocolo MQTT

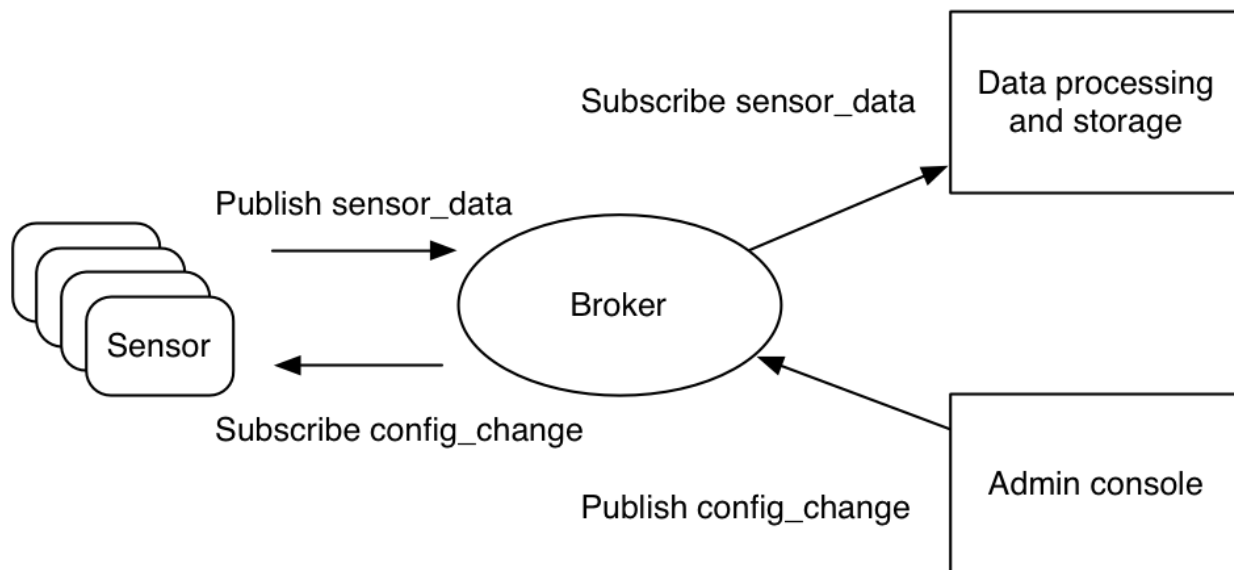
A Internet das Coisas (IoT) revolucionou a interação humana com os dispositivos do dia a dia. De casas inteligentes à automação industrial, a IoT tem o potencial de transformar todos os aspectos de nossas vidas. No entanto, com o grande número de dispositivos em uma rede IoT, a comunicação se torna um grande desafio. É nessa etapa que entra o protocolo MQTT.

MQTT, ou Message Queuing Telemetry Transport, é um protocolo de mensagens leve projetado especificamente para dispositivos IoT. Foi desenvolvido em 1999 por Andy Stanford-Clark da IBM e Arlen Nipper da Arcom para monitorar sistemas de pipeline, que ganhou popularidade no mundo IoT devido à sua simplicidade, eficiência e confiabilidade. Este protocolo foi planejado para enviar dados com precisão sob o longo atraso de rede e condição de rede de baixa largura de banda [MAKWANA, 2017]

No MQTT, os dispositivos se comunicam entre si por meio de um componente central chamado broker. O broker é responsável por receber as mensagens dos publishers e encaminhá-

las aos subscribers. O broker atua como um mediador entre publishers e subscribers, garantindo que as mensagens sejam entregues de forma confiável e eficiente. A arquitetura MQTT pode ser observada na Figura 2 a seguir.

Figura 2 - Fluxograma do funcionamento do MQTT



Fonte: IBM, 2022

Publishers são dispositivos que enviam mensagens para o broker. No MQTT, os publishers podem enviar mensagens para um ou vários tópicos. Os tópicos são sequências hierárquicas que organizam as mensagens em categorias. Por exemplo, um termostato doméstico inteligente pode publicar dados de temperatura em um tópico chamado "casa_inteligente/termostato/temperatura". Os publishers também podem especificar um nível de qualidade de serviço (QoS) para cada mensagem, o que determina a confiabilidade com que a mensagem é entregue aos assinantes (subscribers).

Subscribers são dispositivos que recebem mensagens do broker. No MQTT, os subscribers podem se inscrever em um ou mais tópicos. Quando uma mensagem é publicada em um tópico no qual um assinante está inscrito, o broker encaminha a mensagem para o assinante. Os subscribers também podem especificar um nível de QoS para cada tópico em que se inscrevem, o que determina a confiabilidade com que a mensagem é entregue ao assinante.

Uma das principais vantagens do MQTT é sua característica leve. O protocolo foi projetado para ser eficiente em termos de largura de banda e uso de bateria, tornando-o ideal para dispositivos IoT com recursos limitados. O MQTT usa um tamanho de pacote pequeno e uma sobrecarga baixa, tornando-o muito mais rápido e eficiente do que protocolos tradicionais como o HTTP (BARROS, 2022).

Outra vantagem do MQTT é sua confiabilidade. O protocolo inclui um mecanismo de QoS que permite aos dispositivos especificar o nível de confiabilidade necessário para cada mensagem. Existem três níveis de QoS: QoS0 (no máximo uma vez), QoS1 (pelo menos uma vez) e QoS2 (exatamente uma vez). Quanto maior o nível de QoS, mais confiável é a entrega da mensagem (IBM, 2022).

O modelo de publicação-assinatura usado no MQTT permite uma arquitetura de comunicação altamente flexível e escalável. Os dispositivos podem publicar mensagens em tópicos específicos e outros dispositivos podem se inscrever nesses tópicos para receber mensagens. Isso permite mensagens altamente direcionadas e reduz o tráfego de rede em comparação com a comunicação ponto a ponto tradicional (GOOGLE, 2018).

3 SISTEMA DA ESTAÇÃO METEOROLÓGICA

Uma estação meteorológica é um conjunto de equipamentos e instrumentos que permitem a medição e registro de diversas variáveis climáticas, tais como temperatura, umidade, pressão atmosférica, direção e velocidade do vento, precipitação, entre outras.

Segundo a Organização Meteorológica Mundial (OMM), uma estação meteorológica pode ser definida como "uma instalação fixa ou móvel, na qual são efetuadas observações sistemáticas e regulares das condições meteorológicas em um ou mais pontos da superfície terrestre" (OMM, 2018).

As estações meteorológicas são utilizadas em diversas áreas, como na agricultura, na aviação, na navegação marítima, na previsão do tempo e em estudos climáticos, entre outros. Com base nos dados coletados pelas estações meteorológicas, é possível obter informações importantes para a tomada de decisões em diversas atividades humanas (OMM, 2018).

Além disso, com o avanço da tecnologia, as estações meteorológicas estão se tornando cada vez mais sofisticadas e automatizadas, permitindo a transmissão em tempo real dos dados coletados para centros de monitoramento e análise. Essa evolução tecnológica tem permitido o desenvolvimento de modelos cada vez mais precisos de previsão do tempo e do clima.

3.1 Localização e características gerais

A estação meteorológica se encontra na laje superior no prédio novo do campus da Unesp Sorocaba. Este sistema é composto por quatro equipamentos de coleta de dados climáticos e um painel eletrônico para a recepção e armazenamento desses dados. Além disso, foram empregados componentes de fixação mecânica para assegurar a correta instalação dos

equipamentos. É importante destacar que a manutenção adequada desses equipamentos é fundamental para a obtenção de dados confiáveis e precisos. Sendo assim, faz-se necessário a avaliação constante do sistema meteorológico, a fim de garantir o seu bom funcionamento e, conseqüentemente, a qualidade dos dados coletados.

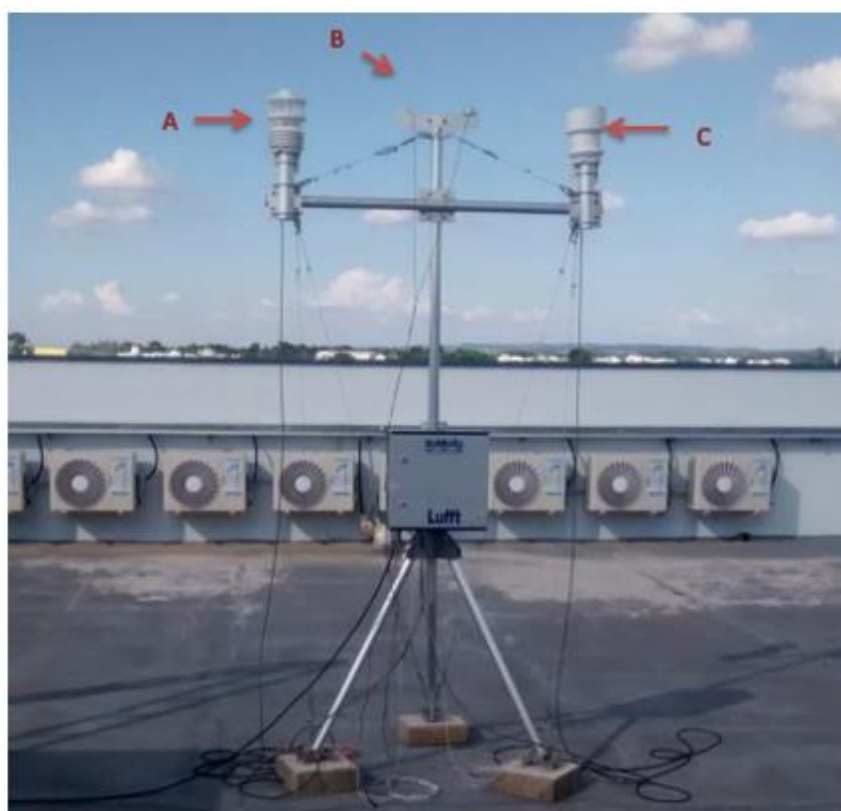
A Figura 3 demonstra a indicação da localização de cada equipamento que compõe a estação, a Tabela 1 apresenta a descrição de cada uma.

Figura 3- Estação Meteorológica e seus sensores.

A - estação compacta

B - piranômetro

C - pluviômetro



Fonte: MARCON, 2016

Tabela 1- Equipamentos da Estação Meteorológica

Componente	Quantidade	Modelo	Fabricante	Função
Estação Compacta	1	WS502-UMB 8375.U10	LUFFT	Sensor climático que realiza as seguintes medições: - Direção do vento - Velocidade do vento - Temperatura do ar - Umidade Relativa - Pressão atmosférica - Radiação
Piranômetro	1	ML-01	EKO	Radiação Global
Pluviômetro	1	WTB100	LUFFT	Mede a precipitação

Fonte: MARCON, 2016

3.2 Estação Compacta

A estação compacta é um equipamento vendido pela empresa RoMiotto que possui sete sensores embarcados que medem a temperatura do ar, pressão atmosférica, umidade relativa, direção do vento, velocidade do vento e radiação global. O modelo instalado é o WS502-UMB 8375.U10, produzido pela empresa alemã Lufft, a linha WS escolhida é uma categoria de estação compacta de baixo custo com uma das mais completas combinações de sensores da marca, uma vez que esse modelo não possui apenas o sensor de precipitação e umidade foliar. A Figura 4 demonstra o equipamento instalado na estação.

Figura 4 - Estação Compacta



Fonte: autoria própria.

É importante realizar o monitoramento da temperatura no terraço do prédio em relação a temperatura da estação climatológica do INMET (Instituto Nacional de Meteorologia), já que a temperatura em terraços tende a ser maior que a atmosférica, devido ao isolamento da manta térmica instalada no chão do local.

Outra medida importante de se observar é a umidade relativa que indica a nebulosidade e possíveis nuvens no céu, pois as condições atmosféricas podem interferir na incidência de radiação solar.

3.3 Pluviômetro

Um pluviômetro é um instrumento usado para medir a quantidade de precipitação, especificamente chuva, em um determinado local durante um determinado período. O instrumento normalmente consiste em um funil ou área de coleta que capta a água da chuva e um recipiente calibrado que mede a quantidade de água coletada. É possível observar o pluviômetro instalado na estação meteorológica na Figura 5 abaixo.

A função de um pluviômetro é fornecer uma medição precisa da quantidade de chuva que ocorreu em um local específico, oferecendo uma resolução de 0,2mm. Esta informação é importante para uma variedade de propósitos, incluindo agricultura, hidrologia e previsão do tempo.

Figura 5 - Pluviômetro



Fonte: autoria própria.

3.4 Piranômetro

Um piranômetro é um tipo de instrumento usado para medir a densidade total do fluxo de radiação solar recebida por uma superfície plana, geralmente expressa em unidades de watts por metro quadrado (W/m^2). É um radiômetro especializado projetado para medir a irradiância solar na faixa de comprimentos de onda do ultravioleta ao infravermelho próximo. O piranômetro instalado na estação meteorológica pode ser observado na Figura 6 a seguir.

A função de medir a radiação solar com um piranômetro é avaliar a quantidade de energia solar disponível em um local e horário específicos. Esta informação é essencial para uma ampla gama de aplicações, incluindo sistemas de energia renovável, projeto de construção e gerenciamento agrícola.

Figura 6 - Piranômetro



Fonte: MARCON, 2016

3.5 Comunicação dos dados

A estação meteorológica possui um datalogger ou coletor de dados do modelo CR800 Series da empresa Campbell Scientific. As principais funcionalidades do equipamento segundo a documentação oficial do CR800 da Campbell Scientific são:

- Múltiplos canais de entrada: a série CR800 pode acomodar uma ampla variedade de entradas de sensor, incluindo interfaces de sensor analógicas, digitais e seriais.
- Registro de dados de alta velocidade: os registradores podem registrar dados em taxas de até 100 Hz, permitindo medições detalhadas de condições ambientais em rápida mudança.
- Programação flexível: Os registradores podem ser programados usando uma variedade de linguagens de programação, como o CRBasic, permitindo a coleta e o processamento de dados personalizáveis.
- Armazenamento de dados: os registradores da série CR800 possuem recursos de armazenamento de dados integrados, com opções para memória interna e dispositivos de armazenamento externo.
- Comunicações: Os registradores suportam uma variedade de opções de comunicação, incluindo Modbus TCP e RTU, Pakbus e interfaces seriais, permitindo transferência de dados flexível e controle remoto.

É importante ressaltar que o protocolo Pakbus já havia sido implementado na estação e essa funcionalidade não poderia ser removida para sua substituição por novos protocolos industriais. Caso houvesse necessidade, seria necessário adicionar o funcionamento de outra forma de comunicação que deve funcionar em conjunto com o protocolo Pakbus.

Para acessar o datalogger CR800 Series é necessário instalar o software PC400 em seu computador. Ele é um pacote de software desenvolvido pela Campbell Scientific que permite aos usuários configurar, programar e baixar dados de seus registradores de dados. O software PC400 inclui uma variedade de ferramentas e recursos para ajudar os usuários a configurar e gerenciar seus sistemas de registro de dados.

Uma das principais ferramentas incluídas no software PC400 é o CRBasic Editor, que permite aos usuários escrever e editar programas de registro de dados personalizados usando a linguagem de programação CRBasic. O editor inclui uma variedade de recursos para facilitar a programação, incluindo realce de sintaxe, preenchimento automático e ferramentas de depuração.

Outra ferramenta útil incluída no software PC400 é o Device Configuration Utility, que permite aos usuários configurar seus registradores de dados e sensores por meio de uma interface gráfica simples. O utilitário inclui opções para definir as configurações do sensor, configurar os parâmetros do registrador de dados e gerenciar as configurações de comunicação.

3.6 Informações transmitidas

O datalogger CR800 Series transmite dados climáticos como a temperatura do ar, umidade relativa, pressão atmosférica, direção do vento, velocidade do vento, precipitação e radiação. A Tabela 2 cita as grandezas físicas das informações transmitidas pela estação meteorológica entregues pela fabricante RoMiotto.

Tabela 2 - Detalhamento das medidas físicas dos sensores da Estação Meteorológica

Parâmetro	Variáveis	Unidade Física	Valores Mínimos	Valores Máximos
Temperatura do ar	PTemp	°C	-50	60
Umidade Relativa	RelHumidity_act	%RH	0	100
Direção do vento	Wind_Direction_act	°(grau)	0	359.9
Velocidade do vento	WindSpeed_act	m/s; km/h; mph	0	75; 270; 167.8
Pressão Atmosférica	RelAirPressure	hPa	300	1200
Precipitação	WTB_Chuva	mm/h	0	200
Radiação	GlobalRadiation_act	W/m ²	0	1400

Fonte: ROMIOTTO, 2016

4 DESENVOLVIMENTO DO SISTEMA EMBARCADO

4.1 Abordagem

Após a pesquisa sobre o sistema da estação meteorológica e avaliação das possíveis alternativas para obter e transmitir os dados coletados, arquitetou-se um sistema embarcado que tornaria a estação comunicável industrialmente atuando como um gateway. Este sistema recebe os dados da estação meteorológica por meio de uma conexão serial e realiza a estruturação desses dados para a integração com a robusta arquitetura IoT do smart campus da UNESP de Sorocaba (PROENÇA, 2022), que coleta e armazena dados de séries temporais com eficiência.

Definiram-se tarefas considerando as condições da estação em 3 etapas: “manutenção e reprogramação da estação meteorológica”, “sistema embarcado”, “visualização”. A topologia da aplicação foi descrita na Figura 7, com etapas de atuação referentes ao ponto “Estação Meteorológica”, “Datalogger” “Sistema embarcado” e desenvolvimento do “Painel”.

Figura 7 - Topologia da aplicação



Fonte: autoria própria.

É importante ressaltar que a comunicação entre o datalogger da estação meteorológica e o sistema embarcado ocorre por comunicação serial UART. Já o envio dos dados do sistema embarcado para o servidor é feito usando o protocolo MQTT. Para o monitoramento das variáveis climáticas em painéis, explorou-se um servidor web com consulta a banco de dados de series temporais.

4.2 Manutenção e reprogramação da Estação Meteorológica

Ao iniciar o projeto, analisou-se as condições em que se encontrava a estação meteorológica, que não estava coletando nenhum dado devido a falha elétrica. Observou-se que diversos pontos deveriam ser corrigidos de forma a garantir um funcionamento com

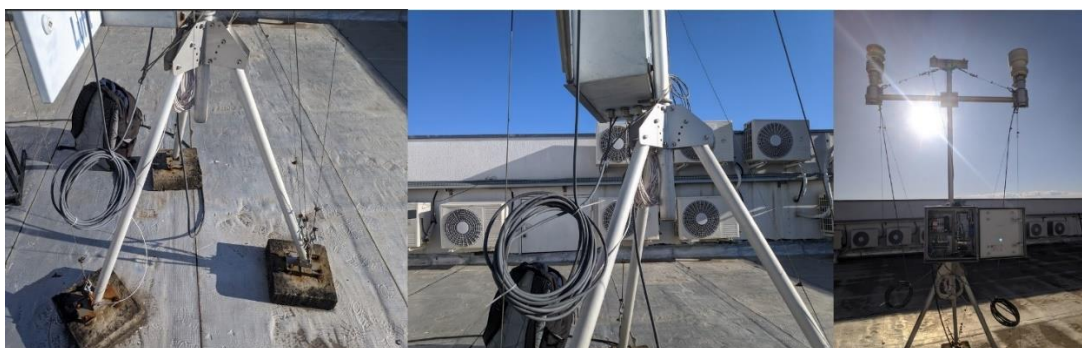
prevenção de riscos futuros. A alimentação da estação meteorológica estava sendo feita em uma pequena sala elétrica conforme a imagem a direita na Figura 8. Para evitar mal contato do filtro de linha, realizou-se sua troca e este foi fixado na parede para uma melhor estética, organização e prevenção de curto-circuito conforme as duas imagens a esquerda da Figura 8. Outro ponto crucial para o avanço do projeto foi a organização dos cabos da estação que, por estarem todos largados no chão e embolados, eram suscetíveis a mal contato e deterioração. A Figura 9 representa a estação após a organização dos cabos. A última manutenção realizada foi a correção do mal contato localizado no painel elétrico de alimentação do sistema da Figura 10, em que o fio de alimentação era facilmente removido sob condições de fortes tempestades ou ventos que movimentavam os fios.

Figura 8 - Painel elétrico do Smart Campus com régua de alimentação fixada na parede.



Fonte: autoria própria.

Figura 9 - Estação após organização dos fios.



Fonte: autoria própria.

Figura 10 - Alimentação da estação meteorológica com emenda de derivação para alimentação do microcontrolador.

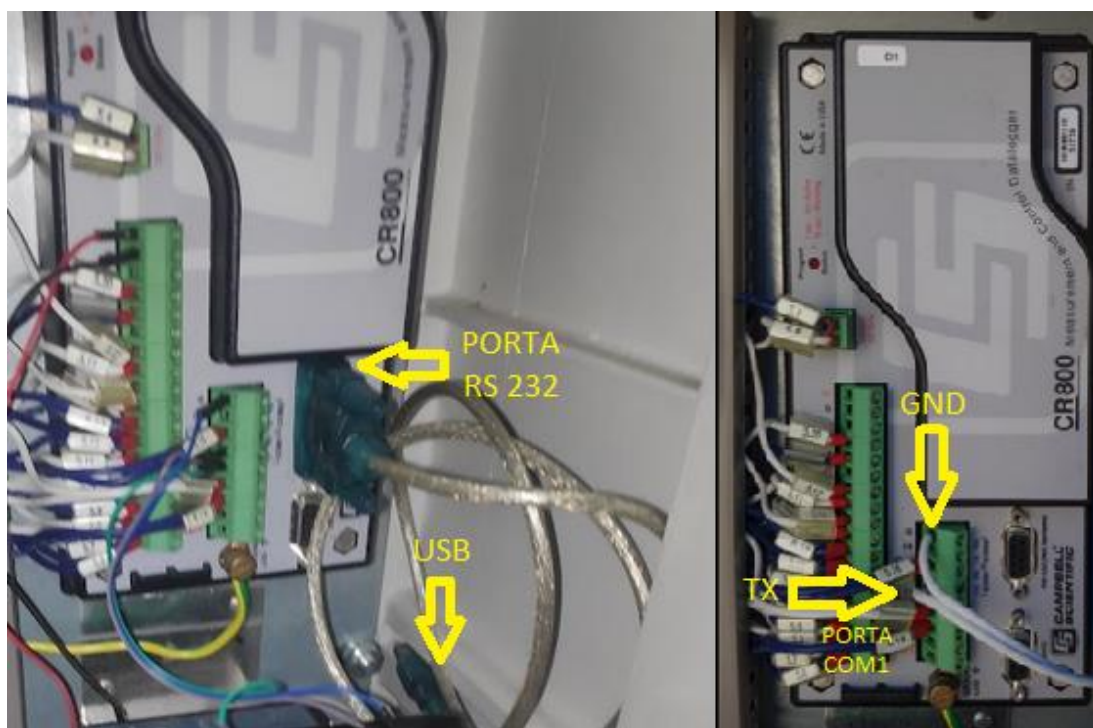


Fonte: autoria própria.

O primeiro passo após a manutenção para o funcionamento da estação, foi a obtenção do código executado pelo datalogger CR800 Series para posterior estudo. Este foi obtido ao conectar o datalogger ao computador pelo PC400 utilizando o conversor da porta RS232 para USB conforme a imagem a esquerda da Figura 11 a seguir. Para extrair o código, é necessário se conectar ao datalogger e apertar o botão de extrair o programa, definido como “Retrieve Program”.

Devido ao requisito dado pelo orientador de manter as conexões existentes e apenas adicionar uma nova comunicação, a porta RS232 estava impossibilitada de ser utilizada no desenvolvimento por já estar sendo utilizada para a comunicação com o protocolo Pakbus. Após uma breve análise, observou-se que a porta COM1 não estava sendo utilizada e era compatível com a comunicação serial. Como é possível observar pela imagem da direita na Figura 11, a porta COM1 é composta pelo conector RX e TX. Dessa forma, foi utilizado apenas a conexão TX da COM1, pois será enviado dados do datalogger ao sistema embarcado pela serial apenas, sem a recepção de dados pelo conector RX. Observa-se também que foi necessário realizar a ligação ao aterramento para o funcionamento correto da serial.

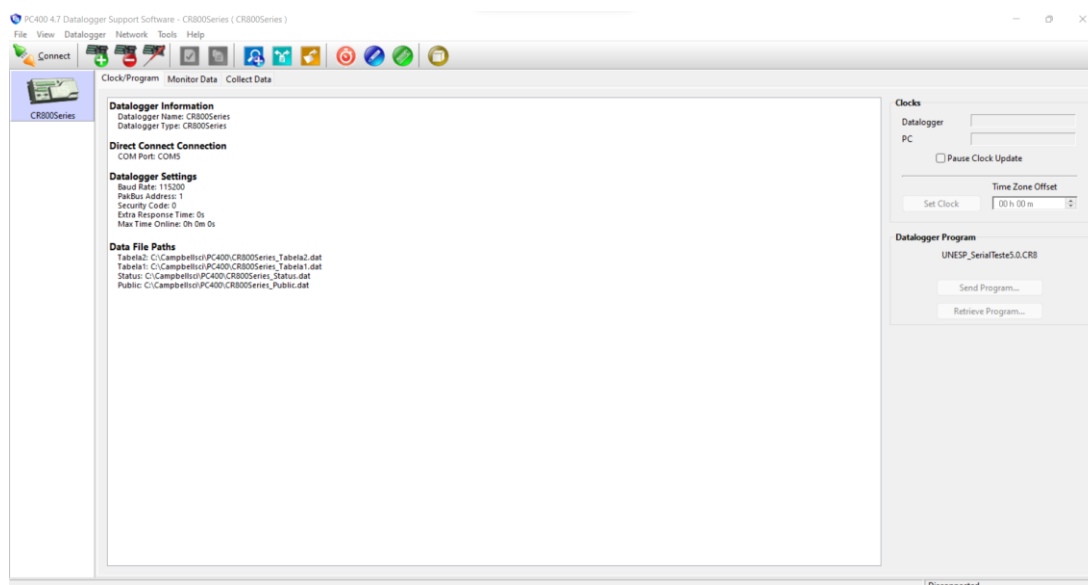
Figura 11 - Porta RS232 e porta COM1 com utilização dos terminais TX e GND.



Fonte: autoria própria.

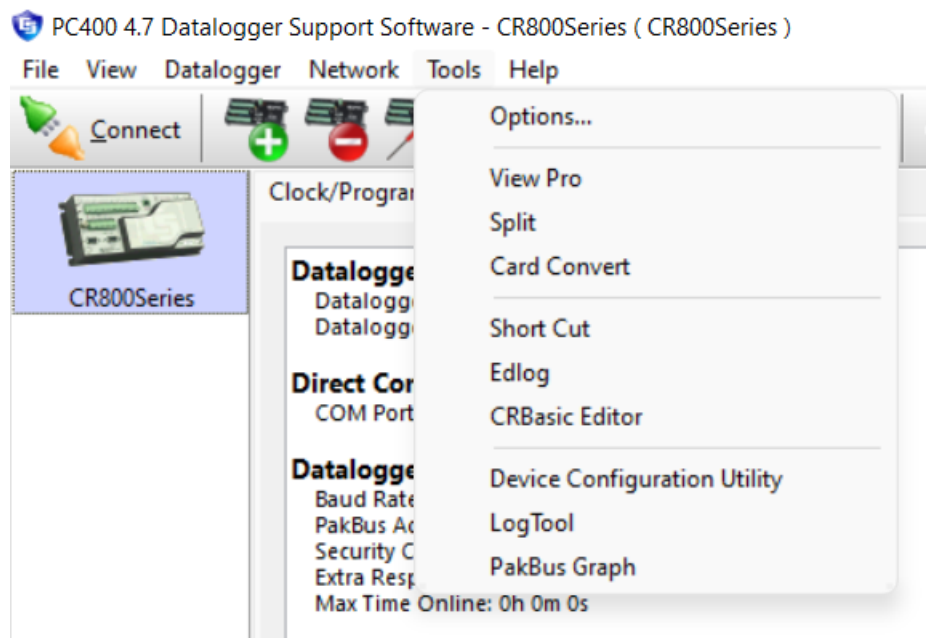
Para a reprogramação do datalogger, foi necessário acessar o software PC400 com o conversor RS232 para USB conforme a Figura 12. Esse software possui diversas ferramentas para a programação dos modelos de datalogger da empresa Campbell Scientific. Um dos softwares utilizados foi o CRBasic Editor, uma IDE de criação de código com ferramentas de auxílio a criação de código na linguagem CRBasic, utilizada no datalogger. O acesso a essa ferramenta de edição de código encontra-se na aba “Tools” conforme a Figura 13.

Figura 12 - Configuração do datalogger CR800 no software PC400.



Fonte: autoria própria.

Figura 13 - Ferramentas nativas do PC400.



Fonte: autoria própria.

Após a execução do software CRBasic Editor, foi aberto o programa que a estação estava executando e foi feita a leitura do código para identificação de seu fluxo de funcionamento, o programa pode ser identificado pela Figura 14.

Figura 14 - Programação do datalogger utilizando o CRBasic Editor.

```

1 'CR800 Series Datalogger
2
3 'To create a different opening program template, type in new
4 'instructions and select Template | Save as Default Template
5 'date:
6 'program author: Cristiano C. da Silva
7
8 'Declare Public Variables
9 'Example:
10 'SequentialMode
11 Public PTemp, batt_volt
12 Public TempContato1          'Pt100 de contato
13 Public HBr1
14 Public HBr2
15 Public TempContato2          'Pt100 de contato
16 Public RS1
17 Public RS2
18 Public RScomp1
19 Public RScomp2
20 Public ML01_Rad
21 Public ML020_Rad
22 Public WTB_Chuva
23 Public Results0(3)
24 Public Results1(2)
25 Public Results2(1)
26 Public Results3(1)
27 Alias Results0(1)=AirTemperature_act
28 Alias Results0(2)=RelHumidity_act
29 Alias Results0(3)=RelAirPressure
30 Alias Results1(2)=GlobalRadiation_act
31 Alias Results2(1)=WindSpeed_act
32 Alias Results3(1)=WindDirection_act
33
34 'Declare a string variable large enough to hold the output string.
35 Public SerialOutString As String * 200
36 'Serial In Declarations
  
```

The right sidebar contains an 'Insert' menu with the following items: #Else, #Elseif, #Endif, #If, #IfDef, &=, *, ^=, /, @, \, ^, ^=, +, +=, <, <<, <=, <>, =, =, >, >=, >>, ABS, AcceptDataRecords, ACDS, ACPower, AddPrecise, Alias, AM25T, AND, AngleDegrees, ApplyandRestartSequence/EndA, ArgosData, ArgosDataRepeat, ArgosError, ArgosSetup, ArgosTransmit, ArrayIndex.

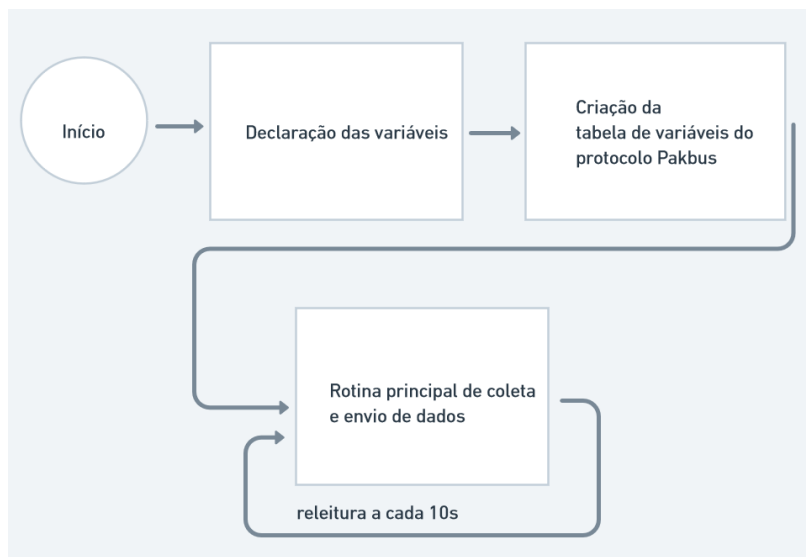
Line: 1 Col: 1 C:\Users\leoue\Desktop\UNESP_SerialTeste5.0.CR8 loaded. Insert

Fonte: autoria própria.

Para um breve entendimento do fluxo de ideias gerou-se um fluxograma disposto na Figura 15. Este programa é responsável por obter o valor das variáveis climáticas enviadas pelos sensores e enviá-los por meio do protocolo Pakbus, um protocolo próprio da empresa Campbell Scientific. Pode-se separar o programa em 3 partes principais, são elas:

- Declaração de variáveis e unidades das variáveis.
- Declaração da tabela de variáveis do protocolo Pakbus.
- Rotina principal de execução para tratamento dos dados e envio das variáveis.

Figura 15 - Fluxograma do funcionamento do código do datalogger.



Fonte: autoria própria.

Com o intuito de se introduzir uma nova forma de comunicação, foi alterado o código original para a adição do envio dos dados pela serial pela porta COM1 conforme a Figura 16 a seguir. O código especificado inicializa a conexão serial com a taxa de transmissão de 115200bps, sem bit de paridade e envia a informação de todas as variáveis juntas concatenadas em uma única informação. O código de envio pela serial foi colocado na função principal do código após a lógica de envio dos dados via Pakbus.

Figura 16 - Adição da lógica de envio de dados pela serial da porta COM1.

```

110 'Serial Out Code
111 'Transmits string "*1:56.789#" out COM1
112 SerialOpen(Com1,115200,19,0,10000) 'Open a serial port
113 'Build the output string
114
115 'Output string via the serial port
116 SerialOut(Com1,"*1:" & PTemp & "#", "",0,100)
117 SerialOut(Com1,"*2:" & batt_volt & "#", "",0,100)
118 SerialOut(Com1,"*3:" & ML01_RAd & "#", "",0,100)
119 SerialOut(Com1,"*4:" & ML020_Rad & "#", "",0,100)
120 SerialOut(Com1,"*5:" & TempContato1 & "#", "",0,100)
121 SerialOut(Com1,"*6:" & TempContato2 & "#", "",0,100)
122 SerialOut(Com1,"*7:" & AirTemperature_act & "#", "",0,100)
123 SerialOut(Com1,"*8:" & RelHumidity_act & "#", "",0,100)
124 SerialOut(Com1,"*9:" & WindSpeed_act & "#", "",0,100)
125 SerialOut(Com1,"*10:" & WindDirection_act & "#", "",0,100)
126 SerialOut(Com1,"*11:" & RelAirPressure & "#", "",0,100)
127 SerialOut(Com1,"*12:" & WindSpeed_act & "#", "",0,100)
128 SerialOut(Com1,"*13:" & HBr1 & "#", "",0,100)
129 SerialOut(Com1,"*14:" & HBr2 & "#", "",0,100)
130 SerialOut(Com1,"*15:" & RS1 & "#", "",0,100)
131 SerialOut(Com1,"*16:" & RS2 & "#", "",0,100)
132 SerialOut(Com1,"*17:" & RScomp1 & "#", "",0,100)
133 SerialOut(Com1,"*18:" & RScomp2 & "#", "",0,100)
134 SerialOut(Com1,"*19:" & WTB_Chuva & "#", "",0,100)
135 SerialOut(Com1,"*20:" & GlobalRadiation_act & "#", "",0,100)
136

```

Fonte: autoria própria.

Um dos principais problemas na atualização do código foi o conflito de portas. Ao realizar o carregamento do código, muitas vezes a serial aparentava estar desconfigurada, apresentando uma taxa de transmissão diferente da especificada no programa resultando na impossibilidade de leitura do dado na serial. Após meses de desenvolvimento, observou-se a existência da ferramenta chamada Device Configuration Utility, já instalada com o PC400, representado pela Figura 17, que habilita a configuração avançada das opções de conexões do datalogger. Após a execução do software, foi possível forçar a taxa de transmissão para 115200bps e também se descobriu uma configuração não utilizada que resultava em um possível conflito com a porta COM1. Após a realização dessas alterações de configuração o código começou a coletar ininterruptamente.

Figura 17 - Configurações avançadas utilizando o Device Configuration Utility.



Fonte: autoria própria.

4.3 Sistema Embarcado

De acordo com o diagrama de bloco presente na figura 12 da sessão 4.1, realizou-se a iteração que diferencia o sinal, faz-se a FFT do resultado, calcula-se o RMS do sinal e armazena-se o resultado da FFT em um vetor. Ao final da iteração é possível obter um vetor com resultados de RMS do espectro de frequência de diferentes estágios de diferenciação. Por fim, identifica-se a posição do menor valor de RMS dentro do vetor, sendo que ela equivale ao tamanho do período do sinal inicial.

Escolheu-se o microcontrolador esp32 wroom 32u para a implementação do projeto por conta de seu custo acessível, alta frequência de operação, capacidade de processamento de dados e alta disponibilidade no mercado. Como o dispositivo deve conseguir se conectar à internet, optou-se pelo uso opcional da antena para o aumento da recepção de sinal de rede, garantindo o envio das informações ao Smart Campus.

A arquitetura do código possui os seguintes requisitos de funcionamento:

- Receber na serial os dados enviados pelo datalogger e adequá-los ao padrão do campus inteligente.
- Se conectar ao WiFi e se reconectar automaticamente.

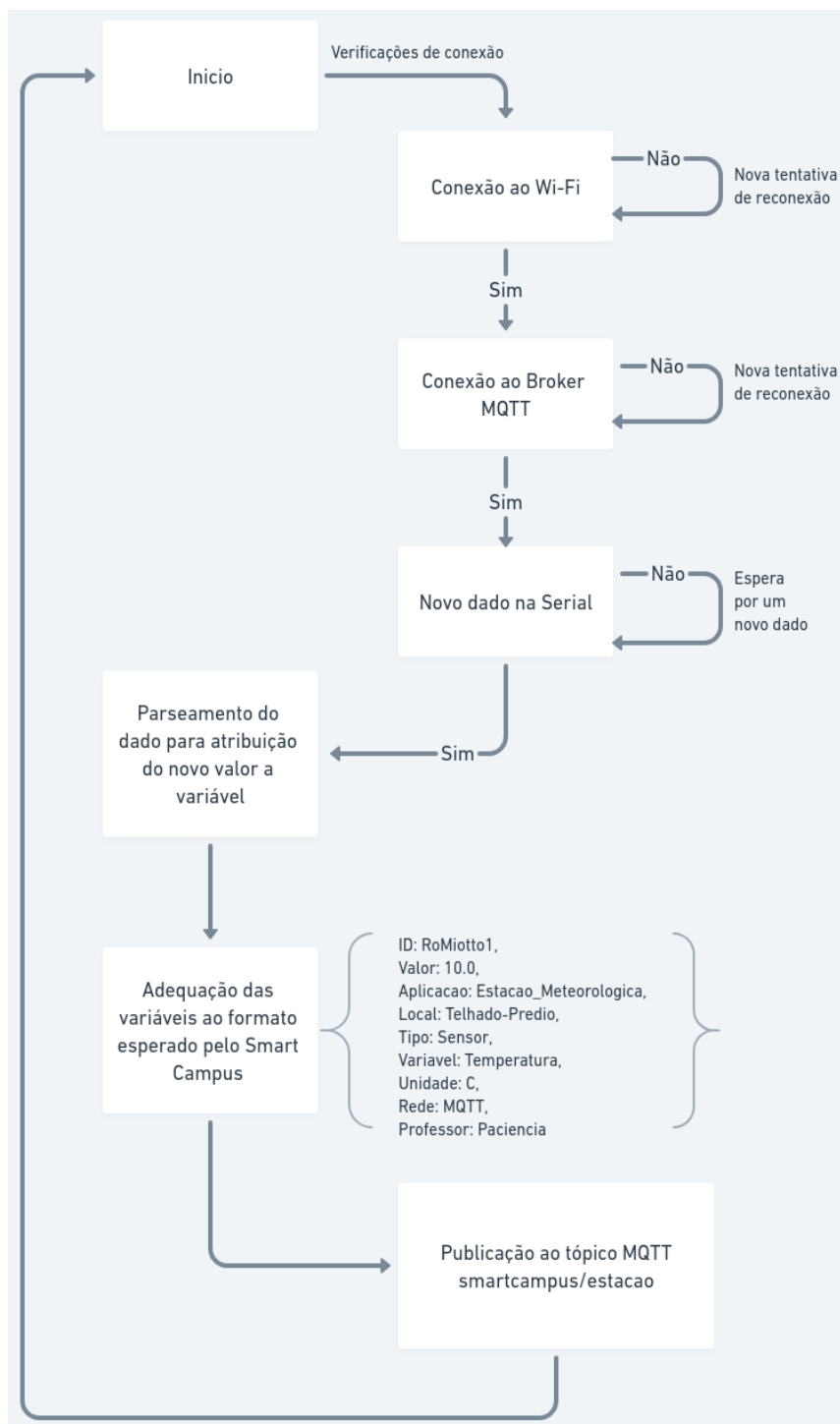
- Deve se conectar e reconectar ao Broker MQTT para envio de mensagens em um tópico.
- Piscar o led a cada nova mensagem recebida e a cada envio de mensagem MQTT.
- Piscar o led caso a mensagem recebida pela serial não siga o padrão esperado.

Para isso, dividiu-se o desenvolvimento do código em etapas, seguindo o requisito de funcionamento.

4.3.1 Fluxo do funcionamento adotado para o código

Para uma melhor visualização do intuito do código, desenhou-se um fluxograma do funcionamento do código do ESP32 representado pela Figura 18 a seguir:

Figura 18 - Fluxograma do código carregado no microcontrolador.



Fonte: autoria própria.

É importante destacar que o programa sempre verifica a existência de conexão com o WiFi e broker MQTT, tentando se reconectar caso ela se perca. Caso tenha sucesso, o sistema espera a chegada de um novo dado para convertê-lo para o formato decimal e alocá-lo a sua

devida variável. Após a realização da leitura corretamente é feito a preparação para o envio das mensagens MQTT no tópico “smartcampus/estacao” utilizando o formato JSON de envio de mensagens com chaves predefinidas pelo sistema Smart Campus.

4.3.2 Estrutura de recebimento das variáveis pela serial

Após filtrar o dado, é realizada a normalização, aplicando-se a normalização com o dado filtrado, obtém-se a série temporal presente na Figura 22.

O Datalogger CR800 Series envia o dado das variáveis da estação meteorológica pela serial a cada 10 segundos com a taxa de transmissão de 115200 bps utilizando o seguinte formato:

*1:24.1308#

Observa-se que o caracter especial “*” identifica o início de uma nova variável, seguido por um sistema chave-valor separados por “:” com a chave incremental entre “1” e “22” e o valor sendo um decimal. Ao final do valor da variável é enviado o caractere “#”. O texto final enviado na serial de acordo com o intervalo estabelecido possui o formato conforme a Figura 19 a seguir, destacando que a informação enviada pela serial é um conjunto de bytes:

Figura 19 - Formato do dado já interpretado enviado pela serial.

```
Novo Dado da Serial!
*1:32.51265#*2:11.89484#*3:926.0274#*4:1583.417#*5:29.1246#*6:29.42377#*7:27.2#*8:63.2#*9:0#*10:0#*11:943.3#*12:0#*13:0.5310287#*14:0.5315032#*15:113.2327#*16:113.4486#*17:111.3327#*18:111.4486#*19:0#*20:992.7#
```

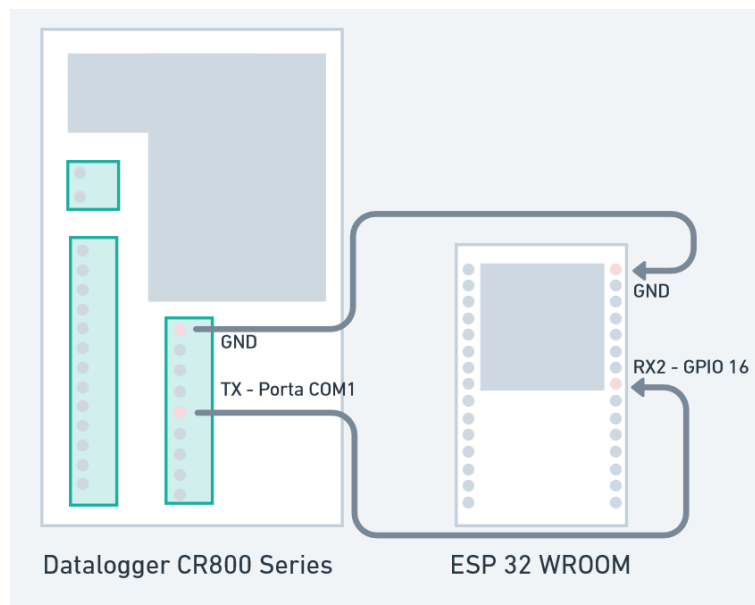
Fonte: autoria própria.

4.3.3 Diagrama de conexão da estação

Após a normalização o dado é encaminhado para a diferenciação iterativa a fim de se encontrar a contagem da iteração que contém a menor RMS da FFT. Executando-se a parte iterativa do algoritmo, foi possível obter o resultado de 144, o qual mostrou-se de acordo com o esperado, discutido na seção 5.1. O algoritmo foi implementado também em um passo anterior a aplicação automática de predição de uma empresa de otimização industrial. O algoritmo foi utilizado em um passo anterior a predição de séries temporais que possuíam caráter periódico, a empresa antes da alteração registrou 467 pedidos de predição, sendo que as predições válidas eram de 351 no mês de setembro de 2021. Após a implementação, no mês de outubro, foram registrados 448 pedidos de predição sendo que 381 foram pedidos válidos, contabilizando um aumento de 9,88% nas predições válidas.

A conexão do datalogger ao ESP32 foi feita por comunicação serial, conectando a saída TX do datalogger a entrada RX2 do ESP32, no pino 16 conforme ilustrado na Figura 20 a seguir.

Figura 20 - Diagrama de conexão serial entre o datalogger e o microcontrolador.



Fonte: autoria própria.

4.3.4 Código .ino do esp32

O código carregado no microcontrolador pode ser dividido em 4 passos principais ordenados por setup, verificações de conexões, manipulação dos dados e envio a arquitetura do campus inteligente.

O primeiro passo é representado pela declaração das bibliotecas utilizadas no desenvolvimento do código seguido pela declaração e atribuição de valor as variáveis utilizadas no projeto, como senhas e portas utilizadas no microcontrolador. Após isso é feita a inicialização das duas portas seriais do controlador utilizando a taxa de transmissão de 115200 bps. É importante ressaltar que foi utilizado duas portas seriais no projeto pois a principal, que possui a entrada com formato micro usb, é destinada a alimentação, carregamento e monitoramento do código, enquanto a segunda porta serial utiliza os pinos 16 e 17 como RX e TX, destinados a leitura dos dados da estação meteorológica. Nesta etapa também é realizado a inicialização do cliente WiFi e MQTT, que serão aplicados no passo seguinte.

Na segunda etapa, é feita a verificação das conexões WiFi, MQTT e serial. Essa verificação é realizada tanto na função “setup” como na função “loop”, de forma a bloquear o prosseguimento da execução do código caso não seja possível realizar a conexão WiFi ou com

o broker MQTT corretamente. Ao se conectar com sucesso, o microcontrolador verifica a chegada de um novo dado na serial para o prosseguimento do próximo passo.

Ao receber um novo dado, que deve chegar conforme o formato estabelecido pelo datalogger, inicia-se o terceiro passo de manipulação de dados. Nesse procedimento é feita a leitura a informação serial recebida e atribuição do valor para sua respectiva variável coletada. Caso o valor da temperatura seja zero, o sistema descarta a leitura e espera pelo próximo dado, ignorando o passo final de envio à arquitetura do campus inteligente.

Ao final do código, é feito o último passo de envio das mensagens ao broker MQTT. Para que o sistema Smart Campus identifique corretamente as informações enviadas, é necessário seguir uma padronização estabelecida no formato JSON conforme a Figura 21 a seguir. É importante ressaltar que para a comunicação MQTT, é necessário enviar uma mensagem por informação de variável no tópico “smartcampus/estacao”. Não é possível enviar muitas mensagens individuais no tópico ao mesmo tempo, sendo necessário o uso de uma função de atraso para criar um intervalo de 200ms entre cada envio.

Figura 21 - Exemplo de estrutura JSON esperada pelo Smart Campus.

```
{
  "ID": "RoMiotto1",
  "Valor": 10.0,
  "Aplicacao": "Estacao_Meteorologica",
  "Local": "Telhado-Predio",
  "Tipo": "Sensor",
  "Variavel": "Temperatura",
  "Unidade": "C",
  "Rede": "MQTT",
  "Professor": "Paciencia"
}
```

Fonte: autoria própria.

4.3.4 Testes de comunicação

Realizou-se uma série de testes para garantir o correto funcionamento do projeto. Para uma melhor organização separou-se os testes de comunicação em três partes, sendo eles, o teste do datalogger, do ESP32 e do Smart Campus.

Na primeira etapa, realizou-se o teste da programação do datalogger. Neste teste, foi garantido que todas as informações das variáveis climáticas estavam sendo enviadas pela serial pela porta COM1 do datalogger a cada dez segundos conforme o padrão estabelecido, citado no item 4.3.2 acima.

Após a consolidação da primeira etapa, testou-se a programação do ESP32. O dispositivo deveria receber o texto completo concatenado de todas as variáveis a cada dez

segundos pela serial pela porta 16 (RX2). Caso o dado da serial esteja fora do formato esperado, o programa deve esperar até a chegada de um novo dado, vale ressaltar que a chegada de um dado fora do padrão indica falha na configuração do datalogger, que pode estar enviando os dados na taxa de transmissão incorreta. Com um novo dado no formato correto, observou se o valor das variáveis, analisando se estas variáveis possuíam os mesmos valores do datalogger, para serem enviadas ao Campus Inteligente por MQTT. Também foi testado se a conexão com o WiFi e o broker MQTT se reestabelecia automaticamente em caso de perda de sinal de rede.

Na última fase do teste, foi observado se os dados chegaram e foram armazenados corretamente no banco de dados do Smart Campus. Para isso, realizou-se consultas no banco de dados InfluxDB para garantir que o dado enviado pelo ESP32 foi salvo corretamente.

4.4 Montagem do Invólucro

4.3.4 Requisitos do invólucro

Com o objetivo de tornar o sistema embarcado mais robusto a choques mecânicos, escolheu-se utilizar um invólucro personalizado, feito com impressora 3D. Para isso foi comprado um modelo próprio para o ESP32 WROOM, com uma boa largura e profundidade, possibilidade de instalação de uma antena externa, encaixe para fixação em Trilho DIN e conectores de fio por pressão conforme ilustrado na Figura 22 a seguir.

Figura 22 - Invólucro feito por impressão 3D.

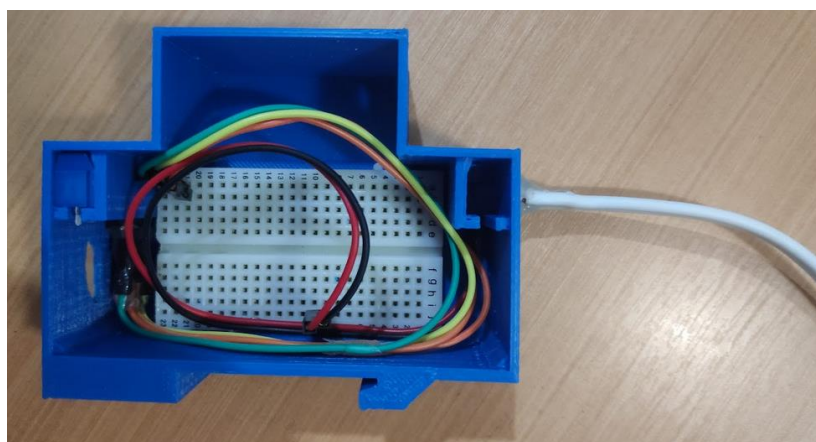


Fonte: autoria própria.

4.4.2 Conexão dos fios

A caixa de proteção necessitava da conexão serial entre o datalogger e o ESP32, conexão de alimentação do microcontrolador e as ligações dos leds de notificação de eventos. Para que exista a possibilidade de modificação da conexão dos fios e até alteração do modelo do microcontrolador, optou-se pela instalação de um protoboard cortado na medida do fundo do invólucro conforme a Figura 23. Desta forma, instalou-se o ESP32 no protoboard e foi feita a ligação dos fios externos de comunicação serial a placa. Também foi ligado um led amarelo e um verde para a sinalização do funcionamento do sistema. Para a alimentação do dispositivo, foi feito um furo extra no invólucro para a passagem do cabo micro usb.

Figura 23 - Vista interna do invólucro com a fixação do protoboard e conexões dos fios.



Fonte: autoria própria.

4.5 Interface Web de Visualização

Foi estabelecido que as representações visuais dos dados coletados pelo datalogger da estação meteorológica seriam distribuídas em duas telas no Grafana, cada uma delas apresentando predominantemente um tipo de formato para a exibição das informações. Cada escopo presente em cada tela pode ser descrito como:

- 1ª tela: Destina-se principalmente a dados instantâneos e à apresentação geral do projeto, devendo apresentar somente informações relevantes de forma responsiva e intuitiva, adaptando-se perfeitamente à tela do usuário.
- 2ª tela: Tem como foco dados históricos obtidos do datalogger ao longo do tempo e também deve ser simples, intuitiva e responsiva ao usuário, de modo a garantir uma apresentação clara e objetiva das informações climáticas.

Estabeleceu-se que na primeira tela estariam dispostas as seguintes informações:

- Apresentação do projeto
- Descrição das variáveis contidas no projeto
- Data de recebimento do dado mais recente
- Imagem da estação meteorológica
- Dado instantâneo do sensor de radiação e irradiação
- Dado instantâneo da radiação global
- Velocidade do vento atual
- Pressão atmosférica atual
- Umidade relativa atual
- Temperatura do ar atual
- Temperatura interna do Painel da Estação atual
- Chuva acumulada desde o último dia
- Chuva acumulada desde o último mês
- Chuva acumulada desde o último ano
- Botão de redirecionamento para a segunda tela

Já para a segunda tela estabeleceu-se os seguintes pontos:

- Botão para voltar para a primeira tela
- Temperatura hoje ao longo do tempo
- Temperatura média por dia ao longo do mês
- Temperatura média por mês ao longo do ano
- Temperatura para o intervalo selecionado na tela
- Radiação Global média por dia ao longo do mês
- Radiação Global média por mês ao longo do ano
- Radiação para o intervalo selecionado na tela
- Irradiação para o intervalo selecionado na tela
- Radiação Global para o intervalo selecionado na tela
- Velocidade e Direção do vento para o intervalo selecionado na tela
- Velocidade do vento para o intervalo selecionado na tela

- Umidade relativa hoje ao longo do tempo
- Umidade relativa média por dia ao longo do mês
- Umidade relativa média por mês ao longo do ano
- Umidade relativa para o intervalo selecionado na tela
- Pressão atmosférica para o intervalo selecionado na tela
- Chuva acumulada por dia ao longo do mês
- Chuva acumulada por mês ao longo do ano

4.5.1 Princípios básicos para as visualizações dos dados

A plataforma web utilizada no Smart Campus é o Grafana. Esse software é uma plataforma de código aberto para visualização e monitoramento de dados que permite aos usuários criar painéis personalizados para visualizar dados de diversas fontes. Ela suporta uma ampla gama de fontes de dados, incluindo bancos de dados, APIs e outras ferramentas de monitoramento, tornando-se uma ferramenta versátil para análise e visualização de dados.

Uma das principais características do Grafana é o suporte a múltiplas fontes de dados, o que permite aos usuários agregar dados de diferentes fontes em um único painel. Essas fontes de dados podem ser configuradas e gerenciadas dentro do Grafana, permitindo que os usuários se conectem a vários bancos de dados, plataformas de nuvem e ferramentas de monitoramento.

Outro aspecto importante do Grafana é o suporte a plugins, que permite aos usuários estender sua funcionalidade e adicionar novos recursos. Os plugins do Grafana podem ser desenvolvidos pela comunidade ou por desenvolvedores terceirizados e podem adicionar novos tipos de visualização, fontes de dados ou integrações com outras ferramentas.

Para criar um painel no Grafana do Smart Campus, o usuário deve selecionar a fonte de dados do InfluxDB e configurá-la com as credenciais e configurações necessárias. Uma vez conectada a fonte de dados, já é possível começar a construir painéis e visualizações para exibir seus dados. Painéis podem ser adicionados ao painel geral e os usuários podem personalizar o layout e a aparência do painel de acordo com suas necessidades, executando consultas ao banco de dados InfluxDB.

O InfluxDB é uma das fontes de dados populares suportadas pelo Grafana, é um banco de dados de séries temporais projetado para armazenar e consultar grandes volumes de dados com marcação de tempo. Em essa implementação do Grafana, utilizou-se a linguagem Flux que tem como objetivo fornecer uma maneira simples e flexível de consultar e manipular dados

de séries temporais, com suporte para várias fontes de dados, tipos de dados e funções de agregação.

4.5.2 Casos especiais obtidos

Foram executadas uma série de consultas especiais ao banco de dados InfluxDB por meio da linguagem Flux para otimizar o uso de processamento do banco de dados de forma a evitar uma sobrecarga no servidor. Observou-se que a quantidade de dados que as consultas buscavam era massiva, de forma a consumir muito recurso de RAM do servidor. Como a taxa de leitura do dado era alta, a consulta retornava uma imensa quantidade de dados a serem renderizadas no painel. Para resolver esse problema, criou-se uma lógica para reduzir a amostragem do conjunto de dados dependendo do intervalo de tempo selecionado no painel conforme exemplificado na Figura 24. Se o intervalo possuir a duração inferior ou equivalente a seis horas, todos os dados desse intervalo são retornados na consulta ao banco de dados para uma determinada variável. Caso contrário é realizado uma redução da frequência de amostragem, agregando dados de um intervalo especificado e retornando a média desses dados. Para customizar o tempo de agregação dos dados foi realizada uma lógica condicional utilizando a duração do intervalo de tempo selecionado no painel como argumento de decisão para cada faixa de intervalo.

Como resultado foram preenchidos os seguintes requisitos:

- Caso o usuário selecione um intervalo de menos de seis horas, todos os dados são apresentados.
- Caso o usuário selecione um intervalo de menos de um dia, é feita uma agregação de dados a cada dez minutos, ou seja, a cada dez minutos de dado, um dado é apresentado.
- Caso o usuário selecione um intervalo de menos de sete dias, é realizada a agregação a cada uma hora.

Caso o intervalo seja acima de sete dias, é feita a agregação a cada quatro horas.

Figura 24 - Consulta com agregação de acordo com o intervalo selecionado na tela.

```

timestart = uint(v: v.timeRangeStart)
timestop = uint(v: v.timeRangeStop)
duration = duration(v:timestop - timestart)

if int(v:duration) <= int(v:6h) then
  from(bucket: "UNESP_SOROCABA_SMART_CAMPUS")
    |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
    |> filter(fn: (r) => r["_measurement"] == "Paciencia")
    |> filter(fn: (r) => r["Aplicacao"] == "Estacao_Meteorologica")
    |> filter(fn: (r) => r["ID"] == "RoMiotto1")
    |> filter(fn: (r) => r["Local"] == "Telhado-Predio")
    |> filter(fn: (r) => r["Variavel"] == "PTemp")
else
  from(bucket: "UNESP_SOROCABA_SMART_CAMPUS")
    |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
    |> filter(fn: (r) => r["_measurement"] == "Paciencia")
    |> filter(fn: (r) => r["Aplicacao"] == "Estacao_Meteorologica")
    |> filter(fn: (r) => r["ID"] == "RoMiotto1")
    |> filter(fn: (r) => r["Local"] == "Telhado-Predio")
    |> filter(fn: (r) => r["Variavel"] == "PTemp")
    |> aggregateWindow(every: if int(v:duration) <= int(v:1d) then 10m else if int(
v:duration) <= int(v:7d) then 1h else 4h, fn: mean)

```

Fonte: autoria própria.

A segunda consulta a seguir realiza uma busca de dados de precipitação do último mês com uma agregação de dados a cada dia, somando esses valores. Essa agregação é feita pela função da linguagem flux chamado `aggregateWindow()` em que é passado como parâmetro o intervalo de agregação dos dados e a forma de agregação. As possíveis formas de agregação incluem a agregação por valor médio, máximo, mínimo e soma dos dados.

Figura 25 - Consulta com agregação de soma de um período de um dia.

```

from(bucket: "UNESP_SOROCABA_SMART_CAMPUS")
|> range(start: -1mo)
|> filter(fn: (r) => r["_measurement"] == "Paciencia")
|> filter(fn: (r) => r["Aplicacao"] == "Estacao_Meteorologica")
|> filter(fn: (r) => r["ID"] == "RoMiotto1")
|> filter(fn: (r) => r["Local"] == "Telhado-Predio")
|> filter(fn: (r) => r["Variavel"] == "WTB_Chuva")
|> aggregateWindow(every: 1d, fn: sum)

```

Fonte: autoria própria.

A última consulta abaixo possui a finalidade de demonstrar o uso da função `last()`, responsável por limitar o retorno da consulta em somente um único dado, sendo este o último dado salvo no banco InfluxDB. A utilização dessa função é essencial para a demonstração de

dados instantâneos pois ela reduz a carga de informações necessárias ao retorno do banco, otimizando o uso de processamento do banco e melhorando a experiência do usuário por tornar a consulta mais rápida e eficiente.

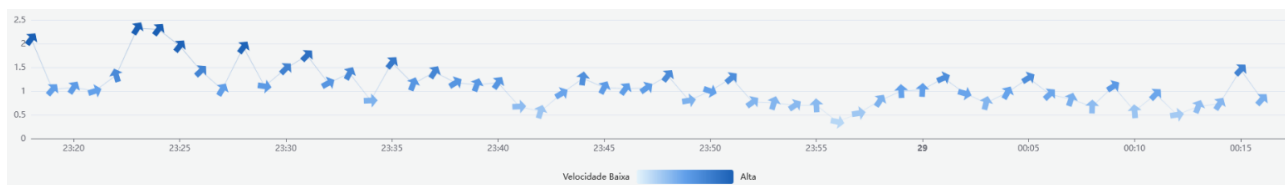
Figura 26 - Consulta com retorno de apenas o último resultado.

```
from(bucket: "UNESP_SOROCABA_SMART_CAMPUS")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "Paciencia")
  |> filter(fn: (r) => r["Aplicacao"] == "Estacao_Meteorologica")
  |> filter(fn: (r) => r["ID"] == "RoMiotto1")
  |> filter(fn: (r) => r["Local"] == "Telhado-Predio")
  |> filter(fn: (r) => r["Variavel"] == "PTemp")
  |> last()
```

Fonte: autoria própria.

Além das consultas personalizadas, utilizou-se um plugin chamado “Apache ECharts” para a montagem de um gráfico personalizado da velocidade e direção do vento, renderizando setas das direções e intensidades do vento em um determinado intervalo, a direção do vento pode ser notada pela mudança do ângulo da seta e a velocidade do vento é dada pela alteração da intensidade da cor da seta e sua altura no eixo Y, conforme a Figura 27.

Figura 27 - Gráfico customizado da velocidade e direção do vento.



Fonte: autoria própria.

Para a montagem desse gráfico, foi necessário buscar os dados relativos à direção do vento e velocidade do vento, que eram salvos em diferentes horários, e gerar uma tabela com as duas variáveis no mesmo horário. Para isso, executou-se uma agregação de dados condicional ao intervalo selecionado na tela, semelhante a primeira consulta exemplificada nessa seção, com a junção dos dados das duas variáveis em uma só tabela. Pode-se observar a consulta e resultado da tabela na Figura 28.

Figura 28 - Consulta com agregação de duas variáveis e tabela retornada com as duas variáveis no mesmo timestamp.

time	wind_direction	wind_speed
2023-05-28 23:19:00	140	1.03
2023-05-28 23:20:00	146	1.08
2023-05-28 23:21:00	108	1.00
2023-05-28 23:22:00	197	1.33
2023-05-28 23:23:00	147	1.22

```

1 timestamp = uint(v: v.timeRangeStart)
2 timestep = uint(v: v.timeRangeStop)
3 duration = duration(v:timestep - timestamp)
4
5
6 query1 = from(bucket: "UNESP_SOROCABA_SMART_CAMPUS")
7   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
8   |> filter(fn: (r) => r["_measurement"] == "Paciencia")
9   |> filter(fn: (r) => r["Aplicacao"] == "Estacao_Meteorologica")
10  |> filter(fn: (r) => r["ID"] == "RoMiotto1")
11  |> filter(fn: (r) => r["Local"] == "Telhado-Predio")
12  |> filter(fn: (r) => r["Variavel"] == "WindSpeed_act")
13  |> aggregateWindow(every: if int(v:duration) <= int(v:2h) then 1m else if int(v:duration) <= int(v:1d) then 10m else if int(v:duration) <= int(v:7d) then 30m else 1h, fn: mean)
14  |> map(fn: (r) => ({ time: r["_time"], wind_speed: r["_value"] })) // select only the columns you need, and rename them as desired
15
16 query2 = from(bucket: "UNESP_SOROCABA_SMART_CAMPUS")
17   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
18   |> filter(fn: (r) => r["_measurement"] == "Paciencia")
19   |> filter(fn: (r) => r["Aplicacao"] == "Estacao_Meteorologica")
20   |> filter(fn: (r) => r["ID"] == "RoMiotto1")
21   |> filter(fn: (r) => r["Local"] == "Telhado-Predio")
22   |> filter(fn: (r) => r["Variavel"] == "WindDirection_act")
23   |> aggregateWindow(every: if int(v:duration) <= int(v:2h) then 1m else if int(v:duration) <= int(v:1d) then 10m else if int(v:duration) <= int(v:7d) then 30m else 1h, fn: mean)
24   |> map(fn: (r) => ({ time: r["_time"], wind_direction: r["_value"] })) // select only the columns you need, and rename them as desired
25
26 join(tables: {query1: query1, query2: query2}, on: ["time"])
27 |> fill(column: "wind_direction", usePrevious: true) // fill null values with the last non-null value
28

```

Fonte: autoria própria.

A Figura 29 demonstra o código na linguagem Javascript responsável pela renderização das setas e correta interpretação dos dados retornados da consulta realizada no InfluxDB. No painel do Apache ECharts é possível renderizar diversos formatos de gráficos. Para a visualização da direção e velocidade do vento, optou-se pela utilização de um gráfico de séries temporais. Dessa forma, no retorno da função, foi retornado um objeto com uma chave denominada “series”, sendo informado ao plugin os valores dos campos de tempo, valor da velocidade do vento e angulo de rotação da seta. Também foi informado na chave do objeto, qual seria o campo do eixo X e eixo Y, sendo eles a velocidade e o tempo, respectivamente. Vale lembrar que para a utilização desse painel, é necessário ter instalado uma versão do Grafana superior a 8.5.

Figura 29 - Código Javascript responsável pela renderização customizada do painel.

```

let wind;
data.series.map((s) => {
  const time = s.fields.find((f) => f.name === "time").values.buffer;
  const value = s.fields.find((f) => f.name === "wind_speed").values.buffer;
  const rotate = s.fields.find((f) => f.name === "wind_direction").values.buffer;

  /**
   Set Wind from 3 arrays
  */
  wind = time.map((id, index) => {
    return { symbolRotate: rotate[index], value: [time[index], value[index]] };
  });
});

return {
  xAxis: {
    type: "time",
  },
  yAxis: {
    type: "value",
  },
  visualMap: {
    orient: "horizontal",
    left: "center",
    min: 0,
    max: 2,
    text: ["Alta", "Velocidade Baixa"],
    dimension: 1,
    inRange: {
      color: ["#e5f4fb", "#62a0ea", "#1a5fb4"],
    },
  },
  series: [
    {
      data: wind,
      type: "line",
      symbol:
        "path://M31 24.7343L21.7917 24.7343V0L9.20826 0L9.20826 24.7343H0L15.5 45L31 24.7343Z",
      symbolSize: 20,
      lineStyle: {
        width: 0.3,
      },
    },
  ],
};

```

Fonte: autoria própria.

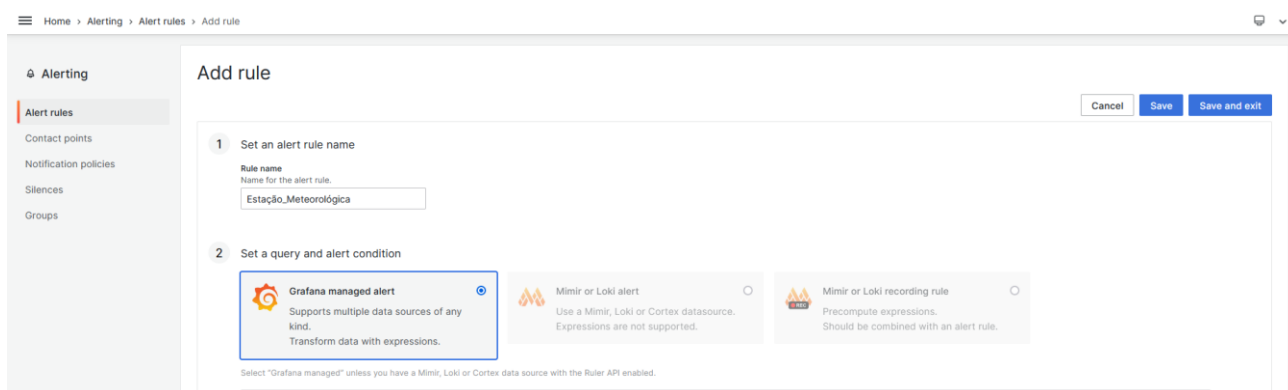
4.6 Configuração de aviso caso o sistema pare de enviar dados

Durante o desenvolvimento deste trabalho, uma das principais preocupações foi assegurar que o administrador do painel pudesse monitorar o funcionamento do sistema embarcado e sua integração com a arquitetura de forma eficiente. Com o intuito de detectar

eventuais problemas no sistema e alertar o usuário em tempo hábil, optou-se por utilizar a funcionalidade de alerta do Grafana. É importante ressaltar que, como a interface do Grafana é a última etapa do processo de coleta e apresentação de dados, qualquer anomalia identificada nessa etapa indica a ocorrência de falhas no sistema como um todo.

No Grafana, é possível escolher diferentes gerenciadores de alerta conforme demonstrado na Figura 30. Como o Smart Campus já utilizava a opção de configuração de alertas por email, escolheu-se essa opção e nomeou-se o alerta como “Estação_Meteorológica”.

Figura 30 - Escolha do nome e plataforma gerenciadora de alertas.



Fonte: autoria própria.

Após a escolha do gerenciador de alertas, configurou-se a lógica de decisão de ativação do alerta. Para o monitoramento do funcionamento da estação meteorológica, escolheu-se monitorar a soma dos valores da variável de temperatura interna do painel, a “PTemp”, durante o período de três horas. Se essa soma for inferior a 1, o alarme será ativado. Optou-se pelo uso da temperatura pois Sorocaba é uma cidade que dificilmente chega em mínimas de temperatura perto de 0 graus. Desta forma, uma soma dos valores nesse intervalo de tempo apontaria uma falha na estação meteorológica. A Figura 31 demonstra a consulta ao banco de dados InfluxDB, que realiza essa soma, seguido de uma definição de um valor limite para a ativação do alerta.

Figura 31 - Lógica de ativação do alerta.

The screenshot displays the Grafana Alerting configuration page. On the left, a sidebar shows navigation options: Alerting, Alert rules, Contact points, Notification policies, Silences, and Groups. The main area is titled 'Alerting' and shows a query editor for the bucket 'UNESP_SOROCABA_SMART_CAMPUS'. The query is a Flux query with the following steps:

```

1 from(bucket: "UNESP_SOROCABA_SMART_CAMPUS")
2   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3   |> filter(fn: (r) => r["measurement"] == "Paciencia")
4   |> filter(fn: (r) => r["Aplicacao"] == "Estacao_Meteorologica")
5   |> filter(fn: (r) => r["ID"] == "R01010101")
6   |> filter(fn: (r) => r["Local"] == "Telhado-Predio")
7   |> filter(fn: (r) => r["Varivel"] == "Ptemp")
8   |> sum()

```

Below the query editor, there are controls for 'Threshold' (Input: A, IS BELOW, 1) and 'Series 1' (0, Normal). A green 'Alert condition' indicator shows '0 firing, 1 normal'. At the bottom, there are buttons for '+ Add query', '+ Add expression', and 'Preview'. A section titled '3 Alert evaluation behavior' is partially visible at the bottom.

Fonte: autoria própria.

Para concluir o cadastro do alerta, definiu-se o intervalo de verificação da consulta da condição para ser realizado a cada trinta minutos, com uma tolerância de três horas para a ativação do alerta. É possível constatar que o Grafana oferece a possibilidade de criar uma mensagem contendo informações sobre possíveis erros, a qual é enviada para uma lista de e-mails previamente cadastrados na interface. Tal funcionalidade permite uma notificação rápida e eficiente dos administradores do sistema. Na Figura 32, apresentam-se as configurações realizadas para o envio dessas mensagens.

Figura 32 - Intervalo de verificação e detalhes do alerta.

The screenshot shows the 'Alert evaluation behavior' configuration page in Grafana. The breadcrumb trail is 'Home > Alerting > Alert rules > Add rule'. The page is divided into two main sections:

3 Alert evaluation behavior

- Folder:** Alerts (selected), estacao_meteorologica (selected)
- Evaluation interval:** Evaluate every 30m
- for:** 3h
- Configure no data and error handling:**
 - Alert state if no data or all values are null: Alerting
 - Alert state if execution error or timeout: Error

4 Add details for your alert rule

Write a summary to help you better manage your alerts. Click here for documentation on how to template annotations and labels.

Summary and annotations:

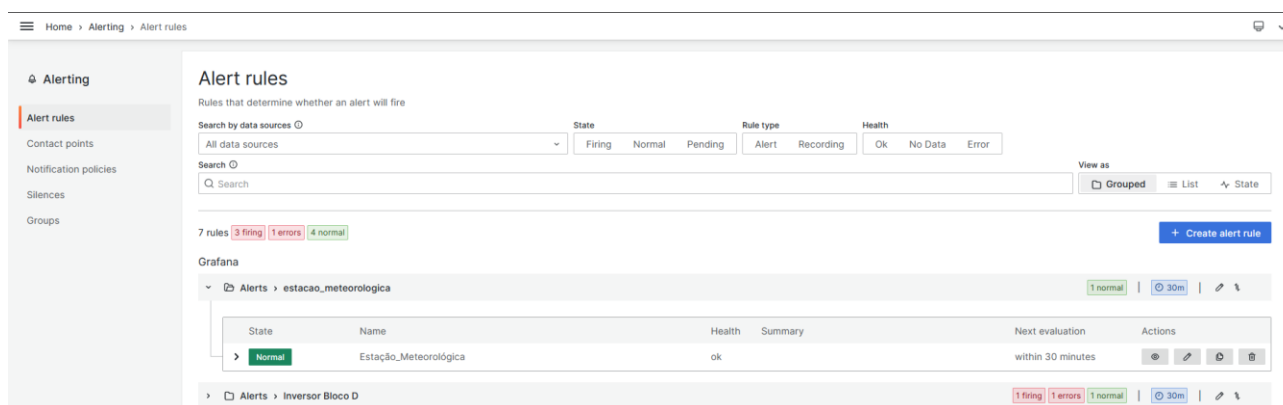
- Summary:** Text
- Description:** dados da estação meteorológica estão ausentes por mais de 3h. Erros comuns: (1) estação pode estar desligada; (2) sinal de internet não pode ser
- Runbook URL:** https://

Buttons at the bottom: + Add annotation, + Set dashboard and panel

Fonte: autoria própria.

Após a conclusão do cadastro do alerta, criado ao apertar o botão salvar no canto superior direito, será possível encontrá-lo na lista de alertas e observar facilmente o intervalo de execução da verificação do alerta e sua saúde. Caso o alerta seja ativado, uma etiqueta vermelha escrito “firing” será renderizada. Se a lógica do alerta resultar em um erro, impossibilitando a verificação do alerta, uma etiqueta vermelha escrito “error” aparecerá. Caso contrário, uma etiqueta verde escrito “normal” será renderizada, indicando que o alerta está funcionando e sua condição não foi ativada. No exemplo da Figura 33, é possível notar que o alerta da estação meteorológica encontra-se funcionando corretamente, com sua execução a cada trinta minutos e a condição do alerta não foi ativada.

Figura 33 - Saúde do alerta cadastrado.



The screenshot displays the 'Alert rules' page in a Grafana interface. The page title is 'Alert rules' and it includes a subtitle 'Rules that determine whether an alert will fire'. There are filters for 'State' (All data sources), 'Rule type' (Alert, Recording), and 'Health' (Ok, No Data, Error). A search bar is present. The main content area shows a summary of 7 rules: 3 firing, 1 error, and 4 normal. Below this, there is a table of alert rules. The first rule is 'Estação_Meteorológica' with a state of 'Normal', health of 'ok', and a next evaluation time of 'within 30 minutes'. The second rule is 'Inversor Bloco D' with a state of '1 firing', 1 error, and 1 normal, and a next evaluation time of '30m'.

State	Name	Health	Summary	Next evaluation	Actions
Normal	Estação_Meteorológica	ok		within 30 minutes	
1 firing, 1 error, 1 normal	Inversor Bloco D			30m	

Fonte: autoria própria

5 RESULTADOS

Devido o envolvimento do projeto englobar diferentes âmbitos, foram subdivididos os resultados em 2 aspectos principais do projeto: os referentes ao sistema embarcado desenvolvido e ao painel montado para a visualização gráfica das informações.

5.1 Sistema Embarcado

O sistema embarcado composto pelo microcontrolador esp32 com antena foi colocado em uma caixa de impressão 3D de estilo industrial, com a conexão da parte superior ao cabo serial e a inferior ligado a fonte de alimentação conforme a Figura 34 abaixo. A implementação do sistema funcionou corretamente, sem que houvesse perda de dados e se reconecta automaticamente caso haja perda de sinal de WiFi ou perda de conexão com o broker MQTT. Caso o dado da serial chegue invalidado, um led amarelo é aceso, indicando visualmente uma falha na comunicação com o datalogger. Há também um led verde que pisca a cada dez segundos quando um novo dado chega ao microcontrolador e este pisca mais vezes seguidas caso o dado seja enviado com sucesso via MQTT. As implementações visuais com os leds facilitam o acompanhamento do status da estação para fornecer uma melhor experiência de uso do sistema caso este precise de manutenção futura. Outro ponto a destacar do projeto é o seu custo, que por utilizar um esp32, foi reduzido significativamente em relação a outras soluções que incluem microprocessadores como o Raspberry. O microcontrolador utilizado trouxe uma maior robustez na qualidade de sinal de comunicação pois apesar do seu baixo custo, ele possui a antena externa que amplifica a cobertura de recepção do sinal, garantindo que o sistema sempre encontre uma rede WiFi para estar conectado.

Figura 34 - Sistema embarcado instalado, apresentado pela caixa azul.



Fonte: autoria própria.

5.2 Painel e interações do usuário

5.2.1 Tela 1 – Informações instantâneas

A tela principal possui o intuito de apresentar o projeto, demonstrando a arquitetura do sistema, suas variáveis e respectivas medidas, apresentando os valores instantâneos das variáveis coletadas. O resultado do design da tela principal pode ser observado da Figura 36 a Figura 38.

Pode-se notar que o início da tela, representada pela Figura 35, tem a responsabilidade de apresentar a gestão de dados da estação meteorológica. A tela explica qual foi o projeto realizado, quais variáveis climáticas compõem o sistema da estação meteorológica, o intuito do sistema desenvolvido e como o dado trafega nessa arquitetura desde a captação do sinal elétrico de cada variável coletada pelo datalogger, passando ao envio dos dados por serial ao sistema embarcado, que repassa as informações por MQTT ao Smart Campus, permitindo a visualização dos dados em um painel do Grafana. Após a introdução do projeto são mostrados

em uma tabela os nomes das variáveis climáticas coletadas pelo sistema, sua descrição e suas respectivas unidades de medida, que podem ser vistas na Figura 36. Já na continuação da tela, representada pela Figura 37, foi fornecido uma visão geral das variáveis de interesse utilizando diferentes painéis do Grafana como o gauge para a rápida identificação da temperatura, com personalização de cor, variando de tons claros para escuros, de acordo com a temperatura atual. Outro painel muito utilizado foi o do tipo stat que disponibiliza numeros de fontes grandes e fundos personalizados, permitindo a visualização clara dos índices atuais de radiação, irradiação solar, radiação global, velocidade do vento, umidade relativa e pressão atmosférica. Com o painel de stat também foi mostrado a data de chegada do último dado, para a finalidade de verificar se a integração desenvolvida está enviando dados para o Smart Campus. Para a contagem da precipitação coletada pelo pluviômetro escolheu-se realizar um painel contendo um stat com maior personalização, nesse caso, realizou-se três consultas diferentes ao banco de dados para obter o volume acumulado de chuva do último dia, mês e ano. Ao final da página foi colocado um botão que redireciona o usuário à página seguinte de informações históricas das variáveis.

Figura 35 - Parte superior da Tela 1.



Fonte: autoria própria.

Figura 36 - Descrição das variáveis climáticas.

Informações sobre as Variáveis Coletadas da Estação Meteorológica					
Variável	PTemp	ML01_Rad	ML020_Rad	AirTemperature_act	RelHumidity_act
Descrição	Temperatura	Sensor de radiação	Sensor de irradiação	Temperatura do ar	Umidade relativa
Unidade	°C	W/m²	Lux	°C	%RH
Variável	WindSpeed_act	Wind_Direction_act	RelAirPressure	WTB_Chuva	GlobalRadiation_act
Descrição	Velocidade do vento	Direção do vento	Pressão atmosférica	Medidor de chuva	Radiação Global
Unidade	W/s	graus de 0 a 360	hPa	mm/h	W/m²

Fonte: autoria própria.

Figura 37 - Parte Inferior da Tela 1.



Fonte: autoria própria.

Com a tela inicial, pode-se observar a diferença entre a temperatura do ar e a interna do painel, que possuem diferentes valores. Isso ocorre pois o painel da estação é uma caixa vedada de metal com componentes eletrônicos que dissipam calor, esquentando mais que o ar conforme a temperatura aumenta. Também pode-se observar a diferença entre a escala e valores da radiação solar descritos pela primeira linha de medidores abaixo do texto com a última data coletada, o sensor de radiação solar (ML01_RAD) e o de radiação global (GlobalRadiation_act) possuem valores próximos e a mesma unidade de medida. Já a variável ML020_Rad possui outra unidade de medida e escala pois ela mede a irradiação solar dada por $\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1} = \text{Lx}$ (Lumen). Ao fim da tela, há um botão de Histórico que redireciona o usuário para a segunda tela, tornando a experiência do sistema mais agradável e fluida.

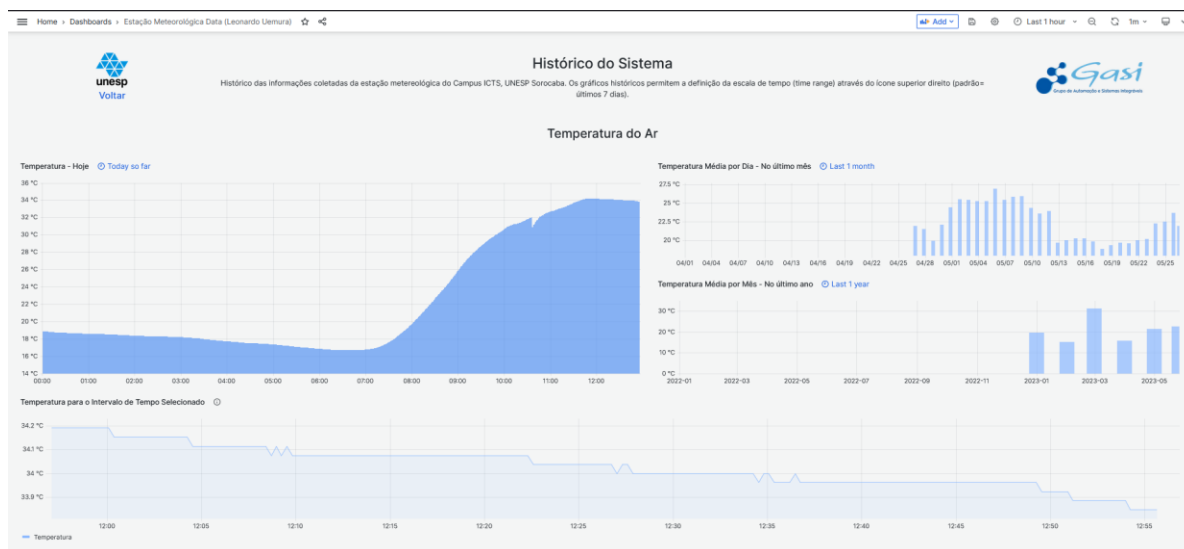
5.2.1 Tela 2 – Informações históricas

A tela de informações históricas possui o intuito de apresentar a série temporal das principais variáveis coletadas para uma análise mais apurada de forma que sejam feitas comparações de correlação de diferentes variáveis e a exportação destes dados para análises de dados utilizando ferramentas externas. O resultado do design da tela de informações históricas pode ser observado da Figura 38 a Figura 42. Para essa tela, escolheu-se dividir a informação em sessões sendo elas “Temperatura do Ar”, “Sensores de Radiação Solar”, “Vento”, “Umidade”, “Pressão” e “Precipitação (Chuva)”.

A Figura 38 demonstra a montagem da sessão de temperatura, com gráficos fixos do histórico de temperatura do dia, temperatura média por dia durante o último mês e temperatura média por mês no último ano. Na parte inferior ainda é possível ver o histórico para um

intervalo customizado selecionado pelo usuário no painel, tornando a tela mais interativa. Essa disposição de informações permite ao usuário observar a mudança da temperatura com diferentes faixas de intervalo e quantidades de pontos, fornecendo toda a liberdade para obtenção de múltiplas formas de percepção da variação da temperatura coletada da estação meteorológica.

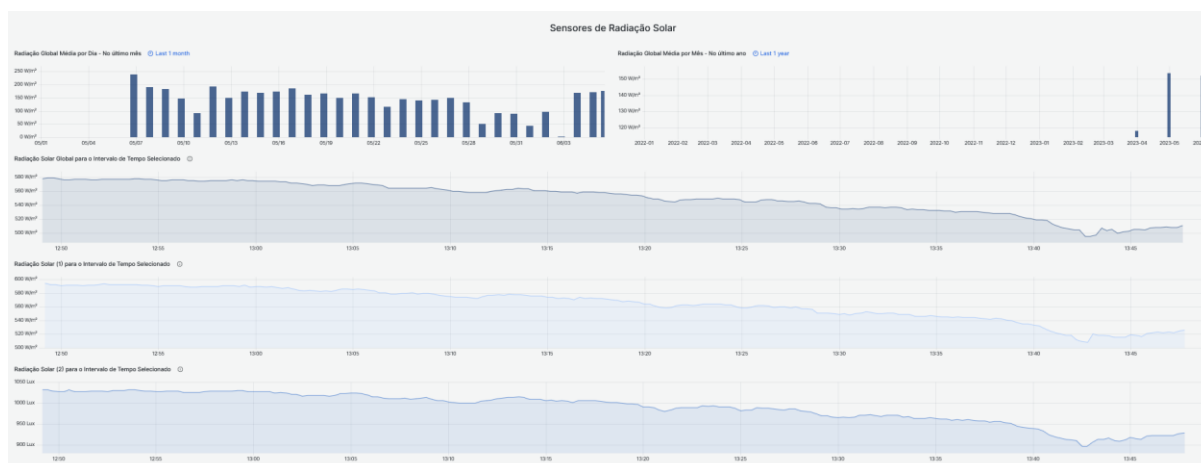
Figura 38 - Dados históricos de temperatura.



Fonte: autoria própria.

A Figura 39 demonstra a montagem da sessão de radiação solar, com gráficos fixos do histórico da média de radiação solar global por dia no último mês e a média de radiação solar global por mês no último ano. Abaixo dos gráficos fixos, inseriu-se gráficos dinâmicos que buscam os dados de radiação solar global, do sensor de radiação e do sensor de irradiação para o intervalo de tempo selecionado na tela no canto superior direito. Com a comparação dos valores das três variáveis de radiação solar, constatou-se que, apesar da unidade de medida do sensor de radiação e irradiação serem diferentes, a única diferença é sua escala de representação destes valores.

Figura 39 - Dados históricos de radiação solar.



Fonte: autoria própria.

A sessão seguinte, representada pela Figura 40, utiliza o gráfico especial que demonstra a direção e velocidade do vento com um formato mais visual, utilizando setas e intensidades de cor, com a busca de dados de acordo com o intervalo de tempo selecionado na tela. Esse gráfico foi o único adotado fora do padrão do Grafana pois a direção do vento é obtida em graus, de zero a 259° , que torna quase impossível a compreensão do dado para usuários leigos. No gráfico de setas, a direção das setas identifica ventos de direções de norte a sul e leste a oeste. As intensidades da cor azul indicam a velocidade do vento, quanto maior a intensidade da cor, maior a velocidade. Também foi criado um gráfico de series temporais da velocidade do vento para o intervalo de tempo selecionado pelo usuário, para a comparação mais detalhada entre os dois formatos de apresentação da velocidade do vento.

Figura 40 - Dados históricos de direção e velocidade do vento.

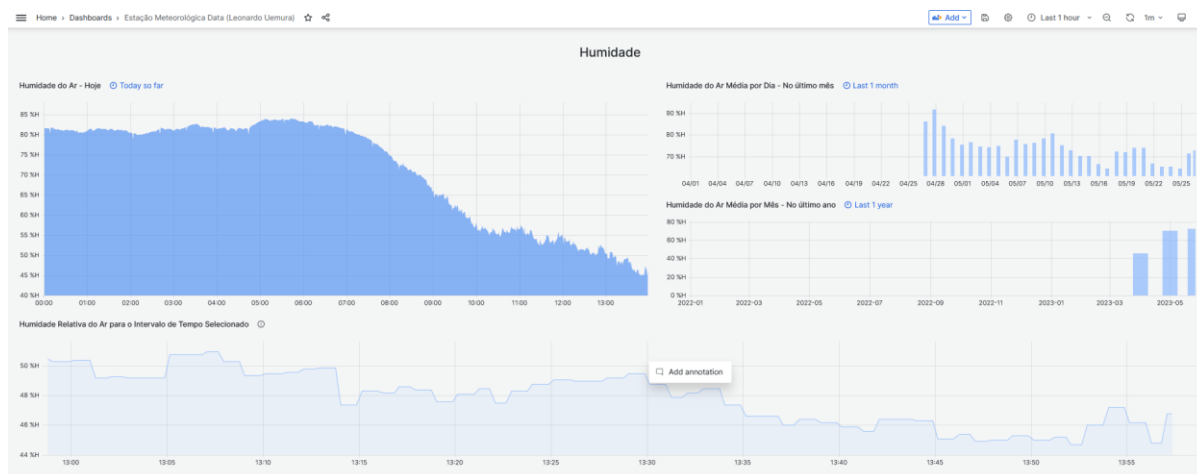


Fonte: autoria própria.

Na Figura 41, a seguir, é exemplificado a sessão de umidade, composta por 3 gráficos fixos e um gráfico dinâmico com a mesma padronização e disposição de informações da sessão de temperatura. É possível observar uma correlação do comportamento da variável de temperatura e umidade, pois com o aumento da temperatura, foi possível constatar a diminuição

da umidade do ar. O monitoramento da umidade é um fator de extrema importância para a saúde e indica riscos de queimada. Também é possível correlacionar a umidade com o desempenho de máquinas, já que a umidade pode afetar a troca de calor de componentes eletrônicos.

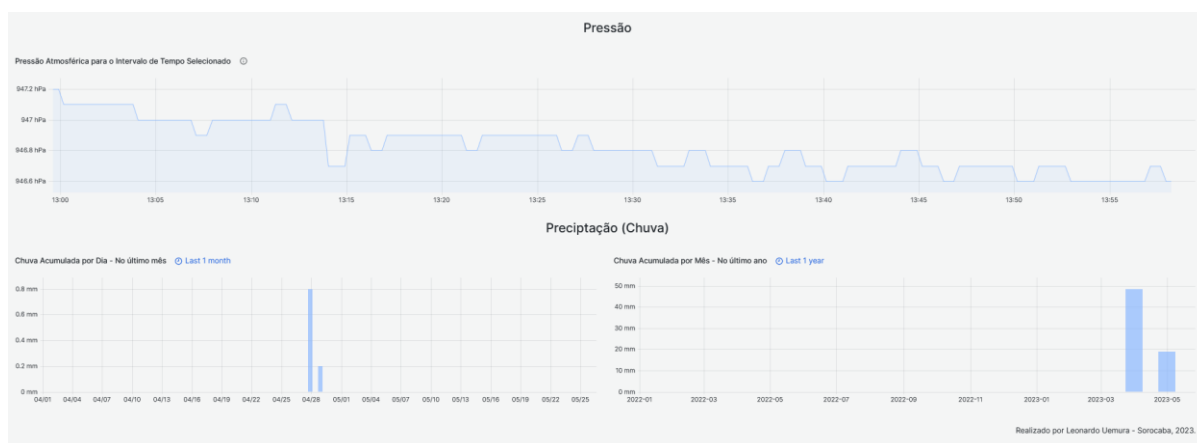
Figura 41 - Dados históricos de Umidade.



Fonte: autoria própria.

Ao final da tela implementou-se o monitoramento da pressão atmosférica para o intervalo de tempo selecionado na tela e a contagem histórica do volume de chuva acumulada por dia no último mês e a chuva acumulada por mês no último ano. Essas duas últimas seções podem ser observadas na Figura 42 abaixo.

Figura 42 - Dados históricos de Pressão e Precipitação.

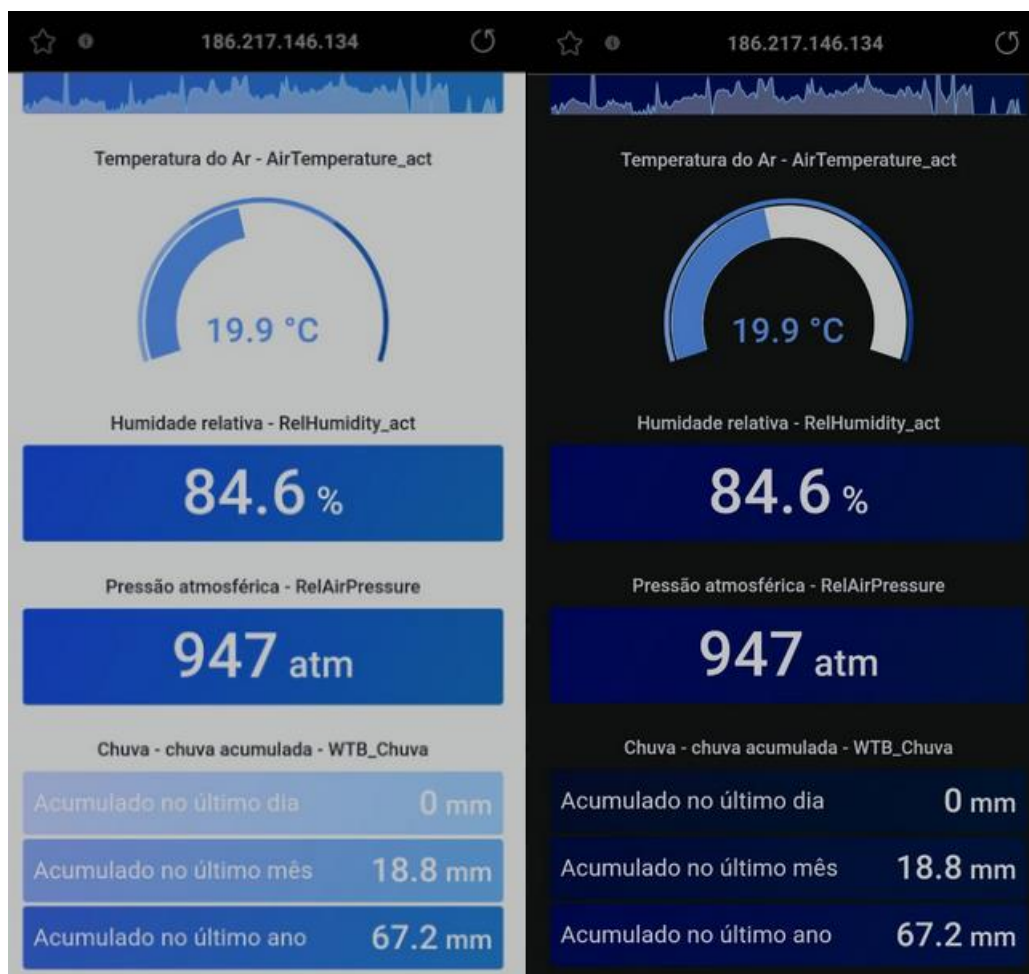


Fonte: autoria própria.

Um dos pontos fortes da construção de painéis web no Grafana é a responsividade da página e adequação a mudança de temas, adequando seu conteúdo para diferentes tamanhos de

tela. A imagem a seguir mostra uma captura de tela de informações históricas feita por um dispositivo móvel utilizando o tema claro e escuro.

Figura 43 - Responsividade mobile para temas claro e escuro.



Fonte: autoria própria.

Pode-se constatar que os painéis construídos possuem um carregamento rápido e responsivo, com valores de tamanho de fácil visualização não importando o dispositivo utilizado. Esse projeto trouxe melhorias significativas de desempenho de processamento em consultas com altos volumes de dados, que pode ser utilizado em implementações futuras. Outro ponto de forte destaque é o formato de disposição das telas divididas pensando em sua usabilidade, com a diferenciação entre telas de variáveis instantâneas e históricas. Caso o usuário tenha a necessidade de extrair os dados para possíveis análises avançadas de correlação de variáveis, é possível utilizar a ferramenta do Grafana de importação de dados para o formato CSV, pois todas as variáveis históricas podem ser vistas com o intervalo de tempo selecionado pelo usuário, tornando-o livre para buscar dados passados sem limitação de tempo.

6 CONCLUSÕES

Com o crescente aumento do uso de equipamentos inteligentes, como dispositivos IoT, surgiu uma nova necessidade que é a coleta e visualização de grandes quantidades de dados. Cada vez mais, o monitoramento de variáveis instantâneas sem salvamento dos dados demonstra uma limitação de possibilidades de percepções de informações. O uso da Internet das Coisas, tem se democratizado, marcando presença como exemplo em domicílios, indústrias e ambientes de saúde. No âmbito acadêmico, é possível encontrar um forte movimento da criação de Smart Campus em faculdades modernas e tecnológicas com aplicações voltadas a gestão de recursos e equipamentos, facilitando o aprendizado acadêmico com o fornecimento de ferramentas de tomada de decisão baseada em evidências para a obtenção de resultados precisos e planejados.

A implementação de um sistema embarcado para a coleta de dados da estação meteorológica para campus inteligente, forneceu mais uma ferramenta para pesquisa e desenvolvimento de futuros projetos acadêmicos na Unesp Sorocaba, agregando valor para toda a comunidade de ambos os cursos, pois fornece uma implementação utilizando conceitos de redes que podem ser utilizados para alunos da Engenharia de Controle e Automação, e disponibiliza dados climáticos que são úteis as análises de alunos de Engenharia Ambiental.

Esse desenvolvimento foi de baixo-custo, comprovando que não é necessário investir tanto dinheiro em hardware para a obtenção da informatização dos dados de um equipamento. Confirmou-se que é possível coletar os dados climáticos da estação meteorológica, enviá-los a um sistema embarcado e armazenar estes dados com precisão no Smart Campus. A utilização de tecnologias modernas de armazenamento de dados com indexação na base do tempo utilizadas na arquitetura do campus inteligente mostrou-se extremamente performática, permitindo o retorno de uma quantidade de dados massiva, com diversas opções de manipulação de dados como a presença das funções de soma, retorno somente do último valor e agregação de dados, que reduz drasticamente o tempo de resposta do banco de dados e o processamento do servidor. A adição dessa arquitetura à escolha da utilização do Grafana para o monitoramento e criação de painéis responsivos e interativos, junto a estratégia de separação das informações em duas telas, por dados instantâneos e históricos, trouxe uma grande melhoria na experiência do usuário, facilitando a compreensão do projeto e funcionamento da estação meteorológica.

A possibilidade de instalação de ferramentas de criação de painéis customizados desenvolvidos por terceiros é outro ponto forte do Smart Campus, que permitiu que os dados

históricos de direção e velocidade do vento fossem visualizados de uma forma simplificada, utilizando a tecnologia para facilitar a compreensão dos dados para pessoas leigas.

Desta forma, o objetivo deste trabalho de desenvolver um sistema embarcado para a coleta e integração de dados de uma estação meteorológica na arquitetura do campus inteligente foi alcançado.

Ainda há melhorias a serem realizadas que podem fomentar o estudo de trabalhos futuros, como a adição dos sensores de temperatura de contato presentes na estação às placas solares, para o monitoramento da temperatura dos painéis fotovoltaicos, criação de análises de ciência de dados com a correlação de variáveis da estação meteorológica e das placas solares e visualização de informações de mínimas e máximas históricas.

REFERÊNCIAS

BANDEIRA, L. K.R. **Smart Campus e a Gestão de Informação**. [2018]. Disponível em: <http://periodicos.ufpb.br/ojs2/index.php/pgc>. Acesso em: 02 jun. de 23.

BARROS, S.A. **Análise de desempenho dos protocolos mqtt**. [2022]. Disponível em: https://repositorio.ufc.br/bitstream/riufc/68365/1/2022_tcc_asbarros.pdf . Acesso em 02 jun. 2023.

CAMPBELL, **CR800 Series Dataloggers**. [2018]. Disponível em: <https://s.campbellsci.com/documents/us/manuals/cr800-cr850.pdf> Acesso em 02 jun. 2023

CLÉTO, Jhonatan. **Projeto de Smart Lock utilizando MicroPython e o Microcontrolador ESP32**. [2021]. Disponível em: https://smartcampus.prefeitura.unicamp.br/pub/artigos_relatorios/Jhonatan_Cleto_Smart_Lock_utilizando_a_ESP32.pdf. Acesso em: 24 mar. 2023.

EMBARCADOS, **O que são sistemas embarcados**. [2023]. Disponível em: <https://embarcados.com.br/o-que-sao-sistemas-embarcados/>. Acesso em: 02 jun. 2023.

ESBRASIL, **Internet das Coisas pode ajudar empresas na automação**. [2023]. Disponível em: <https://esbrasil.com.br/internet-coisas-pode-ajudar-empresas-na-automacao/>. Acesso em: 02 jun. 2023

ESPRESSIF, **Esp32 Wroom**. [2023]. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf. Acesso em 02 jun. 2023

GOOGLE, **MQTT vs HTTP**. [2018]. Disponível em: <http://cloud.google.com/blog/products/iot-devices/http-vs-mqtt-a-tale-of-two-iot-protocols>. Acesso em 02 jun. 2023

IBM DEVELOPER. **IoT MQTT: Why is it good for IoT?** [2017]. Disponível em: <https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/>. Acesso em: 1 jan. 2023.

MAKWANA, Ashwin. **A Survey on MQTT: A Protocol of Internet of Things(IOT)**. [2017]. Disponível em: https://www.researchgate.net/profile/DipaSoni/publication/316018571_A_SURVEY_ON_MQTT_A_PROTOCOL_OF_INTERNET_OF_THINGSIOT/links/58edafd4aca2724f0a26e0bf/A-SURVEY-ON-MQTT-A-PROTOCOL-OF-INTERNET-OF-THINGSIOT.pdf . Acesso em: 20 de nov. de 2022.

MARCON, E. L. **Correlação entre as condições climáticas e atmosféricas com a eficiência de painéis solares na cidade de Sorocaba**. [2016]. Disponível em: <https://bv.fapesp.br/pt/bolsas/154711/correlacao-entre-as-condicoes-climaticas-e-atmosfericas-com-a-eficiencia-de-paineis-solares-na-cidad/> 2016. Acesso em: 14 jun. 2022

MENDES, T. **Iniciativas brasileiras de Internet das Coisas (IoT)**. [2020]. Disponível em: <https://www.napratica.org.br/iniciativas-brasileiras-de-internet-das-coisas-iot/>. Acesso em: 4 mar. 2023.

MORAES, V. O. **Sistema embarcado para a coleta de dados de geração fotovoltaica para Campus Inteligente**. [2023]. Disponível em: <https://repositorio.unesp.br/handle/11449/239575>. Acesso em: 5 mar. 2023.

MQTT.ORG. **MQTT: The Standard for IoT Messaging**. [2021?]. Disponível em: <https://mqtt.org/>. Acesso em: 1 jan. 2023.

NIE, Xiao. **Constructing Smart Campus Based on the Cloud Computing Platform and the Internet of Things**. Atlantis Press. [2013]. Disponível em: <https://www.atlantispress.com/proceedings/iccsee-13/4826>. Acesso em: 25 de out. de 2022.

NITAHARA, A. **Estudo mostra que pandemia intensificou uso das tecnologias digitais**. [2021]. Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2021-11/estudo-mostra-que-pandemia-intensificou-uso-das-tecnologias-digitais>. Acesso em: 12 mai. 2023

OMM, ORGANIZAÇÃO METEOROLÓGICA MUNDIAL. **Manual on Codes**. Volume I.1, Part A. [2018]. Disponível em: https://library.wmo.int/doc_num.php?explnum_id=9965. Acesso em: 4 mar. 2023.

ORACLE. **Internet das Coisas**. [2020]. Disponível em: <https://www.oracle.com/br/internet-of-things/what-is-iot/>. Acesso em: 12 mai. 2023.

PROENÇA, M. H. **Arquitetura IoT para Aplicação em Smart Campus**. [2022]. Disponível em: <https://repositorio.unesp.br/handle/11449/217967>. Acesso em: 5 mar. 2023.

ROMIOTTO, **Estação Compacta**. [2020?]. Disponível em: https://www.romiotto.com.br/index.php?route=product/product/download&pdf=537&document_id=4. Acesso em: 4 fev. 2023

ROMIOTTO, **Produtos**. [2020?]. Disponível em: <http://romiotto.co.br/produtos/>. Acesso em: 4 fev. 2023

SYNERGY, **ESP Wroom 32u Datasheet**. [2017]. Disponível em: https://synergygallery.renesas.com/media/products/135/288/en-US/esp-wroom-32d_esp32-wroom-32u_datasheet_en.pdf. Acesso em: 28 mar. 2023.

TOTVS, **Aplicações da internet das coisas**. [2022]. Disponível em: <https://www.totvs.com/blog/inovacoes/aplicacoes-da-internet-das-coisas/>. Acesso em: 10 jan. 2023

WHITE, E. **Making Embedded Systems**. 1. ed. [2012?] O'Reilly, 2012. v. 1. Acesso em: 30 out. 2022

XIONG, L. **A Study on Smart Campus Model in the Era of Big Data**. 2nd International Conference on Economics, Management Engineering and Education Technology, 2016. Acesso em 4 mar.2023