

# Relatório de Projeto de Sistemas Operacionais Embarcados

Lucas Guimarães Borges  
Faculdade de Ciências e Tecnologia  
Universidade de Brasília

Ryan Augusto Brandão Salles  
Faculdade de Ciências e Tecnologia  
Universidade de Brasília

**Resumo**—Este projeto teve como objetivo o desenvolvimento de um sistema embarcado que visa agir como uma estação meteorológica automática e mais barata que as presentes no mercado, tendo como diferencial prover previsões de condições de temperatura futuras para o usuário final por meio de um modelo de redes neurais. O projeto apresentou como produtos finais um sistema de visualização de dados, dois modelos computacionais para previsão de temperatura, 24 horas e 120 horas e um sistema para obtenção de dados do ambiente com base no sensor BME280. O projeto possui como débitos técnicos a integração entre o modelo e a visualização de dados e uma estrutura mais robusta que seja capaz de evitar a chuva.

## I. INTRODUÇÃO

A agricultura brasileira desempenha um papel estratégico na economia nacional, sendo responsável por aproximadamente 23,2% do Produto Interno Bruto (PIB) em 2024 [1]. No DF, em 2024, segundo dados de Dataviva, cerca de 21% das exportações estaduais foram produtos agrícolas [2]; uma participação econômica significativa, possivelmente apontando que uma otimização do modelo de produção adotado poderia beneficiar não só os produtores, mas o restante da sociedade. A figura 1 apresenta as exportações econômicas do Distrito Federal de forma gráfica para melhor análise.

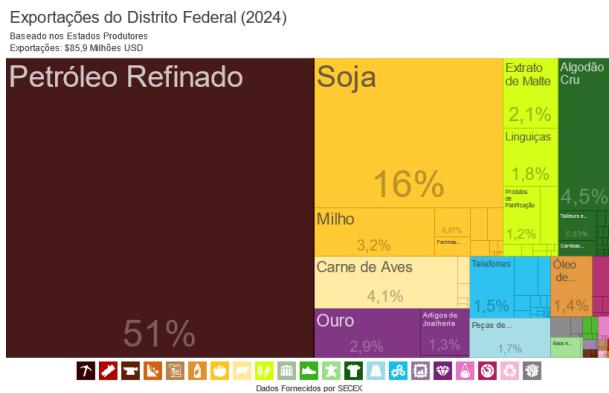


Figura 1. Adaptado de Dataviva (2025) [2].

Segundo Tschiedel e Ferreira, a agricultura de precisão é uma filosofia de gerenciamento de um campo agrícola que busca otimizar a produção utilizando sensores para obter dados sobre o clima e a terra e subsequentemente aumentar a capacidade do administrador de aplicar os insumos de forma correta, ou seja, onde são mais necessários para sustentar a produção [3]. A partir dessa definição, é possível correlacionar

o entendimento do clima de uma determinada região e o correto uso da capacidade sensorial das ferramentas disponíveis.

Uma breve pesquisa acerca das propostas do mercado para a obtenção de leituras meteorológicas locais revela que as soluções de estante tem um grande enfoque na obtenção dos dados, todavia falhando em prover análises mais detalhadas de possíveis mudanças temporais ou emitir alertas para seus usuários, apresentando um vácuo tecnológico capaz de ser explorado. A solução mais comum encontrada pela equipe pode ser exemplificada pelo modelo FT0350 da Gevanti [4], uma estação meteorológica que possui sensores para coleta de dados e um display para apresentação dos dados imediatamente coletados, todavia carece de uma forma de reter esses dados e analisá-los. Soluções similares, com pequenas mudanças na forma de coleta de dados ou visualização, incluem os modelos [5] e [6], fabricados pela AcuRite e Ambient Weather, respectivamente.

Este trabalho apresenta o projeto e execução de um protótipo de sistema computacional integrado com sensores capazes de prover análises simples para seu usuário utilizando técnicas de machine learning, tais como redes neurais. Em outras palavras, este trabalho propõe o desenvolvimento de uma estação meteorológica compacta para os fins de coleta e análise de dados, capaz de comunicar ao usuário final do sistema possíveis alertas e resultados. Além de já demonstrar resultados em outras áreas, segundo a revisão bibliográfica realizada por Benos et. al., redes neurais aplicadas a um ambiente rural parecem ser um tópico em constante progresso [7], permitindo uma oportunidade de estudo de vantagens e dificuldades apresentadas durante o desenvolvimento do sistema pretendido.

Para esse fim, será utilizado o SoC (System on a Chip) Raspberry Pi 3 Modelo B [8] e sensores específicos a serem detalhados posteriormente na lista de materiais, capazes de coletar dados climáticos e ambientais voltados para a aplicação agrícola. Além disso, o sistema incorpora um modelo de inteligência artificial que processa os dados localmente e envia os resultados para um servidor local, possibilitando a visualização em dashboards interativos.



Figura 2. Raspberry Pi 3 Modelo B [8].

## II. SOLUÇÃO PROPOSTA

### A. Descrição de hardware

A fim de obter uma solução mínima, todavia com todas as capacidades necessárias para concluir os objetivos traçados ao longo da introdução, o sensor BME280 foi escolhido com base em suas capacidades e o custo benefício apresentado.

Após uma breve análise da especificação técnica do sensor, foi determinado que ele será capaz de operar nos ambientes propostos, bem como exposto ao ar livre sem apresentar riscos a sua operação. Capaz de comunicação por meio de protocolo I2C ou SPI, o sensor provê leituras de temperatura, umidade e pressão com resolução adequada ao conjunto de dados disponível em domínio público por meio do INMET[9], que disponibiliza suas leituras em ponto flutuante de 2 casas decimais. Ademais, o componente possui documentação suficiente na comunidade de usuários do Raspberry Pi para que seja possível o desenvolvimento do software sem a necessidade de manualmente escrever a interface, sendo meramente necessário conectar o sensor corretamente.

A fim de controlar o sensor, guardar leituras, calcular médias e executar um modelo de aprendizado de máquina, será utilizado o SoC Raspberry Pi 3 Model B (RPI3B) [8], o qual, considerando que foi provido como empréstimo pela instituição, não aparecerá na BOM. O RPI3B possui um processador ARM com 4 núcleos, que serão utilizados para prover poder de multiprocessamento suficiente para sustentar uma solução de software que utilize largamente as capacidades do sistema simultaneamente possibilitando não necessitar de mais periféricos e possibilitando que o usuário utilize um aparelho celular ou computador para monitorar o funcionamento do sistema, contanto que tenha acesso à internet. Em outras palavras, a interface humano-computador será realizada por meio de um site. No mais, as necessidades de armazenamento do sistema são supridas facilmente por um cartão SD, o qual também foi obtido por meio de empréstimo com a instituição.

Em termos de memória provida pelo sistema, o SoC conta com 1Gb de RAM, suficiente para carregar um interpretador Python com modelos de análise computacional periodicamente para prover análises mais detalhadas da

evolução da temperatura com o passar do tempo e/ou carregar o servidor, que será descrito em detalhes na seção de proposta de software.

A figura 3 apresenta um esquemático de circuito de como a conexão foi realizada entre o SoC e o sensor e a tabela I apresenta a Lista de Materiais utilizada pelo projeto.

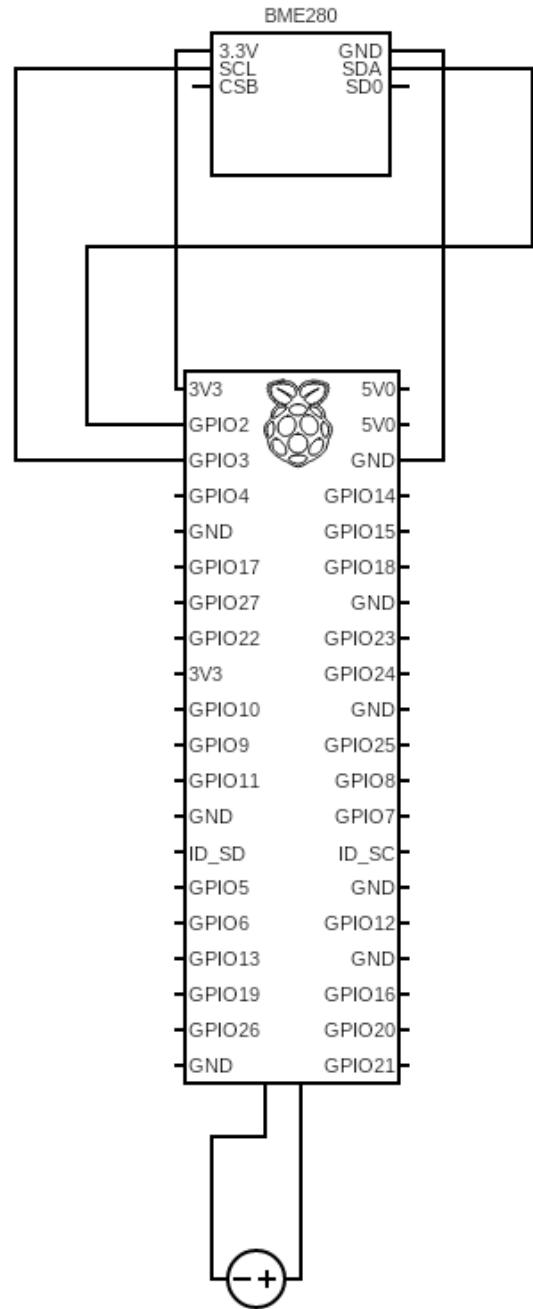


Figura 3. Esquemático de conexão do RPI3B com o sensor BME280

Por fim, a Figura 4 apresenta o diagrama de blocos do sistema, representando os principais componentes de hardware

Tabela I. LISTA DE MATERIAIS.

Componente	Preço unitário	Quantidade
Módulo Sensor BME280	R\$ 52,90	1
Cabo Jumper Fêmea/Fêmea	R\$ 0,20	4
<b>Total</b>	<b>R\$ 53,70</b>	

que serão utilizados para correto e completo funcionamento.

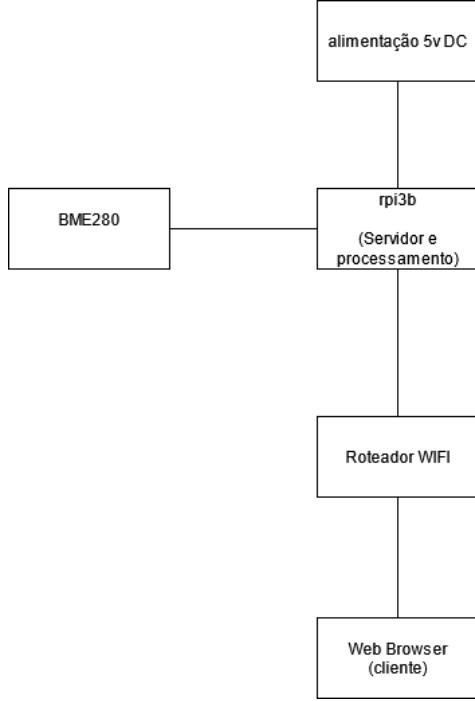


Figura 4. Diagrama de blocos do sistema proposto

### B. Descrição de software

O software funciona como uma estação meteorológica baseada no sensor BME280, integrada a um Raspberry Pi através de comunicação I2C. O processo inicia com a inicialização do barramento e a calibração do sensor, realizando leituras periódicas em uma thread dedicada para não bloquear o servidor web. Os dados coletados passam por validação, sendo posteriormente enviados para um sistema de inteligência artificial em desenvolvimento, responsável por análises preditivas e detecção de padrões climáticos. O sistema mantém os últimos valores em cache, registra logs de eventos e erros, e expõe uma API REST em JSON, facilitando a integração com aplicações externas.

Além disso, conta com um dashboard web desenvolvido em HTML5, CSS e JavaScript, permitindo ao usuário visualizar informações e a coleta de dados. A aplicação é executada como um serviço systemd, configurado para iniciar automaticamente. Por fim, sua arquitetura é baseada em threads, onde a principal executa o servidor Flask responsável pelas requisições HTTP, enquanto uma thread secundária realiza a coleta contínua dos dados do sensor.

A fim de prover um esquema de processamento de dados mais completo, foi proposto um modelo de rede neural recorrente (RNN). Esse tipo de rede possui como vantagem permitir que os valores de saída sejam capazes de afetar os demais.

Por exemplo, um determinado valor de saída 1 poderá afetar o valor 2, o 2, o 3 e assim por diante, tal como apresentado no diagrama da Figura 5. Idealmente, esse modelo de rede neural permitirá que o estado anterior da temperatura seja capaz de afetar os próximos resultados, todavia, caso a performance não esteja como esperada, o modelo será alterado ou descartado em favor de outros tipos, tais como um modelo convolucional.

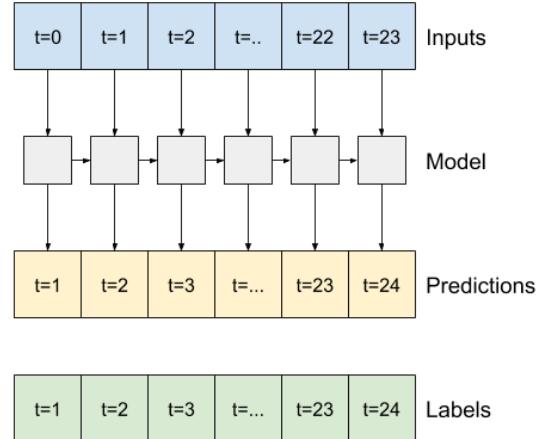


Figura 5. Fonte: adaptado de GOOGLE(2025)[10]

Idealmente, levando em consideração que os dados disponíveis em domínio público possuem granularidade de apenas hora em hora, totalizando 24 medições diárias, é possível que o modelo será ativado para processamento apenas uma vez por hora, permitindo que mais que um modelo utilize o tempo de processamento ocioso.

**1) Coleta de Dados:** A coleta de dados é realizada através de comunicação I2C com o sensor BME280 no Raspberry Pi. A taxa de amostragem é configurável no código-fonte, sendo tipicamente de 60 segundos:

- **Inicialização I2C:** O sistema estabelece comunicação com o BME280 no endereço padrão `0x76`, utilizando a interface I2C da placa.
- **Configuração do sensor:** Aplicação dos parâmetros de operação (modo, oversampling, filtro) definidos na implementação em C++ do módulo `SensorBME280`.
- **Validação da conexão:** Leitura inicial de teste para verificar se o sensor responde corretamente.
- **Loop de coleta:** Laço periódico no binário principal (`main.cpp`) que realiza a leitura dos dados e o registro em arquivo CSV.

**2) Processamento de Dados:** Os dados coletados do BME280 são processados em duas etapas principais: tratamento básico embarcado e processamento por rede neural em ambiente Python:

- **Validação inicial:** Verificação básica de retorno das leituras (checagem de erros de comunicação e valores fora de faixa).

- **Pré-processamento embarcado:** Conversão das leituras brutas para grandezas físicas (temperatura, umidade relativa e pressão) utilizando as rotinas de calibração do BME280 implementadas em C++.
- **Treinamento de modelos:** Os arquivos .csv gerados pela estação são utilizados para tratamento de dados e treinamento de modelos de rede neural em Python (scripts em `tratamento_de_dados/` e `model/python/trainingscript.py`), com armazenamento dos modelos em formato .keras.
- **Inferência com rede neural:** Um servidor de predição em Python (`model/python/prediction_server.py`) carrega os modelos treinados (`t24v1.keras`, `t120v1.keras`) e recebe dados da estação, retornando previsões (por exemplo, séries futuras de temperatura) e permitindo detecção de padrões e tendências climáticas.

#### 3) Armazenamento e Transmissão:

- **Armazenamento em arquivo:** As amostras são registradas em arquivos CSV por meio do módulo CSVLogger, contendo timestamp e variáveis medidas.
- **Serviço HTTP embarcado:** Um servidor HTTP leve, implementado em C++ (`HTTPServer.cpp`), disponibiliza os dados recentes da estação para consumo remoto.
- **Serviço de predição:** O servidor de predição em Python expõe uma API própria para recebimento de dados da estação e devolução das saídas da rede neural, possibilitando integração entre o nó embarcado e o backend de inteligência artificial.
- **Formato de resposta:** As respostas HTTP fornecem os últimos valores coletados e, quando disponível, previsões geradas pelo modelo em formato textual simples (por exemplo, JSON ou `text/plain`), facilitando a integração com aplicações web e scripts externos.

#### 4) Interface com o Usuário:

- **Dashboard web estático:** Uma página web simples em HTML5/CSS/JavaScript (`src/web/index.html`, `script.js`, `style.css`) consome os dados expostos pelo servidor da estação e os apresenta em tempo real.
- **Visualização de medidas e previsões:** Exibição das variáveis de interesse (temperatura, umidade, pressão) e, quando disponível, previsões obtidas pela rede neural, em campos numéricos e/ou gráficos atualizados periodicamente via requisições HTTP.

#### 5) Inserção no Sistema Operacional:

- **Scripts auxiliares:** Scripts shell em `src/BME280/` (por exemplo, `start_server.sh`, `stop_and_clean.sh`, `check_status.sh` e `clear_csv.sh`) facilitam a configuração e a operação do serviço no Raspberry Pi.

#### 6) Arquitetura de Execução:

- **Processo embarcado principal:** Binário em C++ responsável pela leitura periódica do sensor BME280, registro em CSV e atendimento das requisições HTTP de dados brutos.
- **Backend de IA:** Processo em Python dedicado ao carregamento dos modelos de rede neural e à execução de inferências sob demanda, acessado pela estação ou por clientes externos por meio de uma API.
- **Separação de responsabilidades:** A lógica de acesso ao sensor (`SensorBME280`), o registro em arquivo (`CSVLogger`), o atendimento HTTP (`HTTPServer`) e o processamento em rede neural (scripts em `model/python/`) são implementados em módulos distintos, facilitando manutenção e extensões futuras (como inclusão de novos sensores ou novos modelos).

O fluxograma do software descrito pode ser visualizado na figura 6. Os códigos podem ser encontrados no seguinte repositório: [github.com/lcsborges/Trabalho-SOE-2025.2](https://github.com/lcsborges/Trabalho-SOE-2025.2).

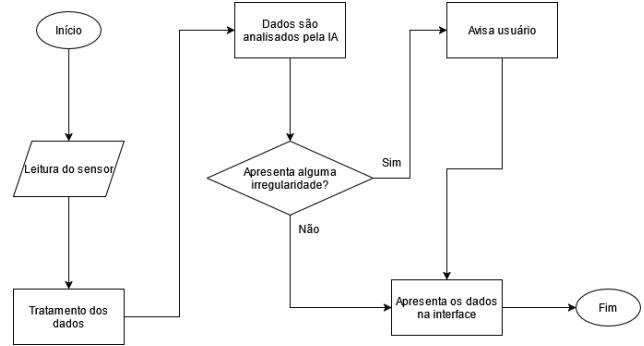


Figura 6. Fluxograma do software

### III. RESULTADOS EXPERIMENTAIS

Essa seção apresenta os experimentos realizados durante o desenvolvimento do projeto com o propósito de validar o funcionamento. Cada experimento possuirá um objetivo bem definido, uma descrição do experimento, uma fase de coleta de dados e um resultado. Cada experimento será contido em sua própria subseção.

#### A. Conexão do sensor BME280

O objetivo deste experimento foi testar a interface entre o sensor utilizado e verificar se a configuração proposta inicialmente seria a correta para realizar leituras utilizando o sensor.

Utilizando a configuração inicialmente proposta na figura 3, a ferramenta i2cdetect corretamente identificou uma conexão em 0x76, como esperado.

Um experimento simples, obtivemos como resultado evidências que a configuração de hardware proposta funcionaria corretamente, apesar de identificado que o conector utilizado facilmente se desconecta do sistema, levando a inconsistências. Realizar uma conexão por meio de uma solda possivelmente resolveria o problema observado.

### *B. Coleta de dados em tempos pré definidos I*

O foco desse experimento foi assegurar que o sistema operacional possuia as capacidades necessárias para coletar leituras em tempos precisos, além de comprovar efetivamente que seria possível coletar os dados e comunicar com o sensor de forma programática.

Utilizando a biblioteca wiringpi para leitura, foi possível realizar a fase de leitura e tratamento dos dados obtidos. Apesar de não uma solução sofisticada, um simples sleep foi utilizado para retardar as leituras do sensor.

O experimento, portanto, foi um sucesso. As leituras de temperatura, pressão e umidade condizem com as esperadas para o tempo do Distrito Federal.

### *C. Permanência e recuperação de dados coletados*

A fim de garantir que os dados coletados estariam disponíveis para comparação histórica e visualização por parte do usuário e para uso no treinamento de modelos computacionais, esse experimento teve como objetivo comprovar que os dados coletados poderiam ser permanecidos em um arquivo dentro do sistema operacional da Raspberry Pi. Mais especificamente, um arquivo .csv, levado em conta o formato do conjunto de dados utilizado para treinamento do modelo.

Para cumprir esse objetivo, foi elaborado uma pequena seção de código que fica responsável por escrever em um arquivo entre as etapas de leitura e espera.

O experimento foi um sucesso parcial. O código elaborado foi capaz de recuperar os dados até certo ponto, depois de alguns minutos parava de puxar os dados do arquivo .csv e precisa ser investigado.

### *D. Visualização I*

Esse experimento foi realizado visando permitir que o usuário seja capaz de acompanhar as leituras do sensor. Para isso, uma página web de visualização hospedada dentro da Raspberry foi elaborada.

A página elaborada foi capaz de corretamente acessar os dados guardados em .csv e, portanto, esse experimento foi um sucesso. Os dados agora são atualizados sem que o usuário precise recarregar as páginas.

### *E. Avaliação do uso de recursos pelo treinamento do modelo II*

Esse experimento foi elaborado a fim de observar o uso de recursos computacionais por parte do treinamento. Dependendo do uso de recursos observados, a plataforma de treinamento do modelo deve ser ajustada de acordo, geralmente exigindo uma GPU compatível com a plataforma Nvidia CUDA para acelerar o treinamento com sucesso.

Para tal, foi elaborado um script de treinamento para um modelo que toma como entrada 24 temperaturas instantâneas das últimas 24 horas e retorna como saída uma previsão das próximas 24 temperaturas horárias. A expectativa seria que

o sistema necessaria de hardware especial para completar os treinamentos em tempo hábil. Surpreendentemente, pela simplicidade do modelo, que possui apenas 3 camadas, foi capaz de completar uma sessão de treinamento em pouco menos de um minuto. O script foi escrito em Python e executou em uma máquina CPU padrão do Google Colab, permitindo até 85 horas de execução ininterrupta. Fosse o caso da necessidade de utilizar uma das máquinas com uma GPU, seria necessário concluir qualquer atividade necessária para salvar o modelo em até 2 horas.

Esse experimento apresentou indícios da possibilidade de concluir um treinamento utilizando a própria Raspberry. Experimentos posteriores serão necessários para dizer com maior precisão o uso de processamento e memória durante o treinamento do modelo. Como subproduto, também gerou o primeiro modelo desse trabalho, o t24v1. O experimento foi um sucesso parcial ao permitir noção do uso de recursos.

### *F. Avaliação do uso de recursos pelo treinamento do modelo II*

Após o sucesso parcial dos experimentos apresentados na subseção III-E, a conclusão lógica seria avaliar o modelo aumentando a quantidade de dados utilizados de 1 dia para 5 dias para previsão de 5 dias de temperaturas instantâneas, mínimas e máximas. Para tal, um segundo script de tratamento de dados foi elaborado, juntamente com a aquisição de 5 anos de dados por meio do INMET, dos anos de 2020 a 2024. Os dados, após as operações de preenchimento de dados nulos de radiação e interpolação de demais chaves foram agrupados no conjunto de dados nomeado dataset2.csv. O treinamento consistiu de 20 sessões de treinamento pelos dados disponíveis e apresentou um sucesso parcial ao obter apenas cerca de 50% de precisão ao final, gerando como subproduto o modelo t120v1, que utiliza a arquitetura de Rede Neural Convolucional. O modelo obtido utiliza 120 triplas de floats para realização das previsões e devolve as próximas 120 triplas esperadas.

Durante o treinamento, o uso de recursos de processamento foi mais expressivo, demorando cerca de 15 segundos por sessão para conclusão do processamento. Dito isso, o modelo segue utilizando cerca de 1.5 Gb de memória RAM para treinamento sem nenhuma técnica de otimização de recursos de memória. Apesar de menos promissor para possibilidades de treinamento diretamente na raspberry pi, ao menos indica que, para essa tarefa, as possíveis próximas versões dos modelos não precisarão de hardware especializado para treinamento.

### *G. Avaliação da precisão dos modelos após treinamento*

Este experimento teve como objetivo verificar a precisão dos modelos treinados com base em dados do inmet para a região do Distrito Federal. Para tal, após treinamento dos modelos, testes foram realizados para avaliar sua performance com base em uma parcela dos dados que foram propositalmente separados apenas para esse propósito, ou seja, o modelo não teria visto esses dados durante o treinamento.

Como apresenta a Figura 7, o modelo de 24 horas foi capaz de aproximar a curva de temperaturas mínimas e máximas

razoavelmente, apesar de apresentar claras dificuldades de identificar a progressão de temperaturas instantâneas, o que seria aceitável não fosse o caso de erros consideráveis em temperaturas mínimas e máximas horárias. O modelo aparenta ter alguma dificuldade de aproximar para curvas mais rebuscadas, aproximando para o que parecem ser curvas senoidais.

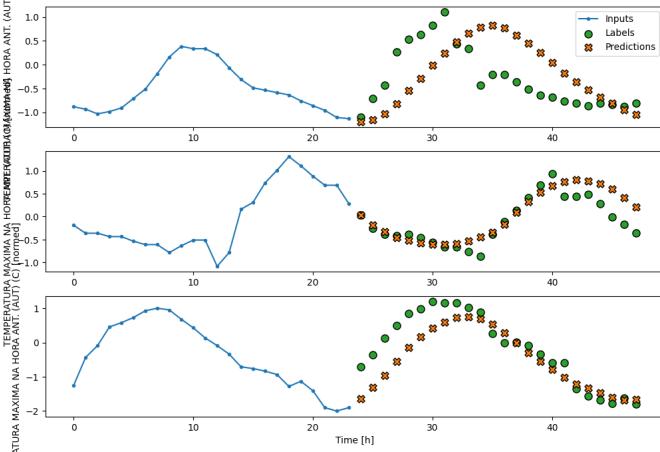


Figura 7. Análise de predição do modelo de 24 horas

O modelo de 120 horas apresentou um problema similar ao modelo de 24 horas, apenas replicando a forma da curva do primeiro dia para os demais períodos da predição em determinados casos de teste, tal como apresenta a Figura 8.

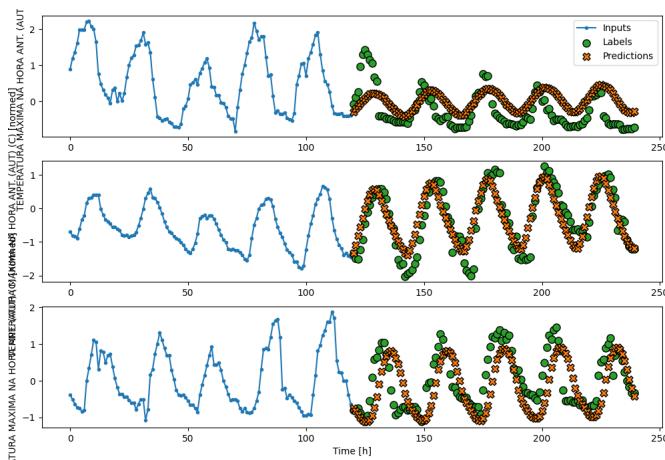


Figura 8. Análise de predição do modelo de 120 horas

Tais resultados podem indicar a necessidade de uma arquitetura de rede diferente ou, inclusive, apontar os limites da arquitetura utilizada que, apesar de consumir poucos recursos computacionais, deixou a desejar em precisão.

#### *H. Avaliação de performance de predição do modelo na raspberry pi*

Esse experimento teve como objetivo avaliar a possibilidade de utilizar a funcionalidade de predição do modelo diretamente na placa raspberry pi. O modelo escolhido para

realizar a predição foi o t120v1 e o conjunto de dados foi o dataset2.

Para a realização do experimento, um script de predição foi elaborado, o qual utiliza 120 triplas de temperaturas instantâneas, mínimas e máximas do início do conjunto de dados e as agrupa num vetor com as dimensões (1,120,3).

Apesar de dificuldades relacionadas às dependências necessárias para executar o script e dificuldades com a biblioteca TensorFlow, após ajustes de configuração, o modelo foi capaz de predizer as próximas 120 triplas demorando cerca de 2 minutos para execução e utilizando 220 Mb da memória disponível na placa.

O experimento foi um sucesso ao executar completamente a predição em paralelo com as demais atividades de responsabilidade da placa, como servir o site de acesso aos dados e obter leituras do sensor. O uso de recursos indica a possibilidade que mais de um modelo computacional poderia estar presente na memória.

#### *I. Estrutura*

Este experimento teve como objetivo avaliar a possibilidade de uma estrutura que protegeria a raspberry durante seu funcionamento.

A estrutura foi elaborada a partir de uma embalagem de shampoo com alguns cortes, permitindo acessar as portas para encaixe de alimentação da energia e comunicação via um cabo LAN, protegendo parcialmente a placa, como apresenta a Figura 9.

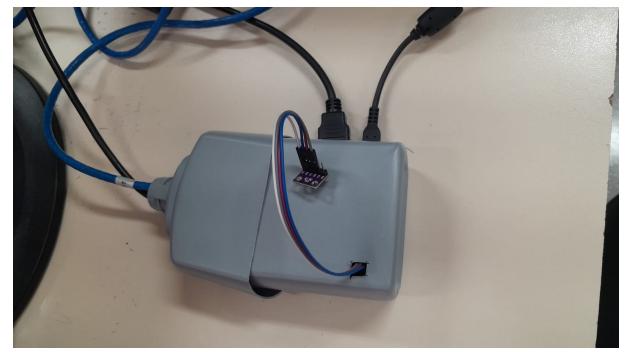


Figura 9. Estrutura Montada

A estrutura contém três partes: um corpo, uma tampa e uma placa, onde a raspberry fica presa, como apresenta a Figura 10.



Figura 10. Partes da Estrutura

Apesar de improvisada, a estrutura foi capaz de proteger a raspberry durante o funcionamento em um ambiente fechado. Demais experimentos serão necessários para desenvolver uma estrutura que funcione também em um ambiente aberto.

#### IV. CONCLUSÕES

Em análise com base no objetivo global, criar um sistema embarcado capaz de coletar dados meteorológicos e predizer temperaturas para auxílio na área da agricultura do cerrado, ou seja, uma pequena estação meteorológica que automaticamente analisa os dados coletados, o sistema foi um sucesso parcial, apresentando dificuldades de integração entre seus componentes de software e precisão de apenas cerca de 50% para previsão de temperaturas futuras. Ademais, o produto final apresenta débitos técnicos que devem ser corrigidos para melhor atender o usuário.

A fim de medir corretamente temperaturas, o modelo utilizou o sensor MPU para obter dados meteorológicos, comunicando por uma interface i2c e controlado por software, capaz de guardar temperaturas em formato .csv. Para visualização dos dados coletados, foi desenvolvido um pequeno site hosteado localmente no sistema. Esse componente apresenta dois débitos técnicos principais: a incapacidade de gerar médias horárias e a falta de integração total com as previsões do modelo de aprendizado de máquina.

Os modelos finais utilizados para testagem dos dados obtidos, o t24v1 e t120v1, apresentam problemas quanto ao treinamento. Ambos são incapazes de obter precisão maior que cerca de 50% e o modelo de 5 dias, o t120v1, é incapaz de corretamente atualizar temperaturas máximas que variam com o passar dos dias, tornando-o não melhor que o modelo de 24 horas na atividade de previsão, apesar de consumir mais recursos para treinamento e utilização. A utilização do modelo dentro da placa não consumiu demasiados recursos, todavia, apesar de não indicado na documentação, exigiu adaptações técnicas das funções de chamada.

A interface de usuário final apresentou dificuldades de integração com o modelo computacional devido a constrições

temporais. Apesar de não um desafio técnico enorme, não foi possível garantir o seu funcionamento dentro do prazo esperado para entrega e essa etapa tornou-se um débito técnico a ser desenvolvido em demais projetos.

A estrutura apresentada não possui garantia de funcionamento em ambientes abertos, o que limita os casos de uso onde poderia ser utilizada.

Por fim, apesar da disponibilidade de outros tipos de dados, como pressão e umidade, tais dados não foram utilizados na análise por parte do modelo, apesar de apresentados ao usuário. A inserção de tais dados dentro da análise realizada pelo modelo computacional apresenta a possibilidade para um modelo mais preciso e que seja capaz de prover uma análise mais interessante que apenas temperaturas de um ponto de vista da agricultura.

É possível comentar algum aprendizado obtido devido ao projeto. O desenvolvimento de um sistema embarcado é razoavelmente desafiador e exige um planejamento mais robusto que o adotado. É necessária uma certa disciplina para cumprir com os prazos adotados. Se propor a executar um projeto que é aparentemente simples exige do executor separar uma quantidade de tempo que foi absolutamente subestimada por parte de pelo menos um dos autores.

Em conclusão, próximas etapas para evolução de um possível subsequente projeto com base no apresentado seriam:

- Estudo de arquiteturas para um modelo computacional mais preciso.
- Integração de componentes de software ainda não integrados, mais especificamente a interface de visualização com a análise por parte do modelo.
- Desenvolvimento de um tratamento de dados para extração de temperaturas mínimas e máximas horárias.
- Inserção de demais informações disponíveis na análise dos dados.
- Desenvolvimento de uma estrutura mais robusta, permitindo outros locais de funcionamento para o sistema.

#### AGRADECIMENTOS

Os autores gostariam de agradecer os Drs. Diogo C. Garcia e Vinicius C. Rispoli pelo auxílio com o desenvolvimento dos modelos t24v1 e t120v1.

#### REFERÊNCIAS

- [1] C. da Agricultura e Pecuária do Brasil (CNA) e Cepea/USP. (2025, Apr.) Pib do agronegócio fecha 2024 com crescimento de 1,81%. [Online]. Available: <https://www.cnabrasil.org.br/noticias/pib-do-agronegocio-fecha-2024-com-crescimento-de-1-81>
- [2] DataViva – Governo de Minas Gerais / Ministério da Economia. (2025) Imports/exports – tree map. [Online]. Available: [https://www.dataviva.info/pt/build\\_graph/secex/3df/all/all?view=Imports/Exports&graph=tree\\_map](https://www.dataviva.info/pt/build_graph/secex/3df/all/all?view=Imports/Exports&graph=tree_map)

- [3] M. Tschiedel and M. F. Ferreira, "Introdução à agricultura de precisão: conceitos e vantagens," *Ciência Rural*, vol. 32, no. 1, p. 159–163, Feb 2002. [Online]. Available: <https://doi.org/10.1590/S0103-84782002000100027>
- [4] Gevanti, "Gevanti Estação Meteorológica Sem Fio, Sensor 9 Em 1 E Display Lcd," Disponível em <https://www.amazon.com.br/Gevanti-Esta%C3%A7%C3%A3o-Meteorol%C3%B3gica-Sensor-Display/dp/B0C6F74VLG> (31/10/2025).
- [5] AcuRite, "AcuRite Estação meteorológica profissional Iris (5 em 1) com visor LCD e detecção de raios (01022M)," Disponível em <https://www.amazon.com.br/Estao-01022M-Relmpago-Temperatura-Deteco/dp/B01MEFGGMF> (31/10/2025).
- [6] Ambient Weather, "Ambient Weather Estação meteorológica inteligente WS-2902 WiFi," Disponível em <https://www.amazon.com.br/meteorol%C3%B3gica-inteligente-WS-2902B-Ambient-Weather/dp/B01N5TEHLI> (31/10/2025).
- [7] L. Benos, A. C. Tagarakis, G. Dolias, R. Berruto, D. Kateris, and D. Bochtis, "Machine learning in agriculture: A comprehensive updated review," *Sensors*, vol. 21, no. 11, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/11/3758>
- [8] Raspberry Pi Foundation, "Raspberry Pi 3 Model B," Disponível em <https://www.raspberrypi.com/products/raspberry-pi-3-model-b/> (04/09/2025).
- [9] I. N. de Meteorologia. (2025, Oct.) Página principal. [Online]. Available: <https://portal.inmet.gov.br/>
- [10] Google. (2025) Time series forecasting. [Online]. Available: [https://www.tensorflow.org/tutorials/structured\\_data/time\\_series](https://www.tensorflow.org/tutorials/structured_data/time_series)