



INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DE PERNAMBUCO

Campus Garanhuns

Tecnologia em Análise e Desenvolvimento de Sistemas

ELIANE DE MELO CORDEIRO

**AUTOMAÇÃO DO MANEJO TÉRMICO EM SUINOCULTURA: uma abordagem
baseada em sistemas embarcados e web.**

Garanhuns - PE

2026

ELIANE DE MELO CORDEIRO

**AUTOMAÇÃO DO MANEJO TÉRMICO EM SUINOCULTURA: uma abordagem
baseada em sistemas embarcados e web.**

Trabalho de conclusão de curso apresentado à
Coordenação do Curso Superior de Tecnologia
em Análise e Desenvolvimento de Sistemas do
Instituto Federal de Ciência e Tecnologia
Campus Garanhuns, como requisito para
obtenção do título de Tecnólogo em Análise e
Desenvolvimento de Sistemas.

Orientador: Prof. Ms. Ivson Galdino Borges.

Garanhuns - PE

2026

C794

Cordeiro, Eliane de Melo

Automação do manejo térmico em suinocultura: uma abordagem baseada em sistemas embarcados e web / Eliane de Melo Cordeiro ; orientador Ivson Galdino Borges, 2026.

61f. : il.

Orientador: Ivson Galdino Borges.

Trabalho de Conclusão de Curso (Graduação) – Instituto Federal de Pernambuco. Pró-Reitoria de Ensino. Diretoria de Ensino. Campus Garanhuns. Coordenação do Curso de Tecnólogo em Análise e Desenvolvimento de Sistemas. Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, 2026.

1. Sistemas embarcados. 2. Automação. 3. Aplicações web. I. Título. II. Borges, Ivson Galdino (orientador). III. Instituto Federal de Pernambuco.

CDD

004.16

Louise Machado Freire Dias –

CRB4/2267

ELIANE DE MELO CORDEIRO

**AUTOMAÇÃO DO MANEJO TÉRMICO EM SUINOCULTURA: uma abordagem
baseada em sistemas embarcados e web.**

Trabalho de conclusão de curso apresentado à
Coordenação do Curso Superior de Tecnologia
em Análise e Desenvolvimento de Sistemas do
Instituto Federal de Ciência e Tecnologia
Campus Garanhuns, como requisito para
obtenção do título de Tecnólogo em Análise e
Desenvolvimento de Sistemas.

Garanhuns, 16 de Julho de 2026.

Prof. Ms. Ivson Galdino Borges

Prof. Ms. Luís Eduardo Tenório Silva

Prof. Dr. Leonardo Soares e Silva

Garanhuns - PE

2026

AGRADECIMENTOS

Primeiramente, gostaria de agradecer ao meu orientador, Prof. Ms. Ivson Galdino Borges, pela confiança depositada em mim e por toda orientação ao longo do desenvolvimento deste trabalho.

Agradeço também ao corpo docente do Instituto Federal de Pernambuco *campus* Garanhuns, que desempenha um papel indispensável na formação acadêmica dos estudantes e se esforça diariamente para entregar qualidade e excelência à educação pública.

Este trabalho representa, além do encerramento de um ciclo acadêmico, uma conquista pessoal por muito tempo almejada. Devo isso à minha avó, Leonora Rocha Wanderley, que partiu há 7 anos, mas continua viva em mim e que sem ela nada disso seria possível. Agradeço pelo seu carinho, seu amor e pela saudade que sinto, pois ela é prova que tive a grande felicidade de ter a melhor vó do mundo.

RESUMO

A suinocultura desempenha um papel fundamental na economia do Brasil, sendo responsável pela geração de emprego e renda. Em Pernambuco, o cenário não é diferente: apesar do estado possuir uma economia diversificada e o setor ser descentralizado entre as regiões, diversas iniciativas buscam impulsionar essa produção, composta majoritariamente por pequenos e médios produtores. O ciclo produtivo exige rigoroso controle do bem-estar animal para garantir a saúde e o alto desempenho do rebanho. Nesse contexto, o conforto térmico destaca-se como um fator crítico para o desenvolvimento adequado dos suínos. Visando automatizar a climatização das instalações, o presente trabalho propõe o desenvolvimento de um sistema de manejo térmico que integra um módulo embarcado a uma plataforma *web*. O *hardware* atua localmente na granja, utilizando sensores de temperatura para acionar ou desativar atuadores de forma autônoma. Simultaneamente, a aplicação *web* funciona como um painel de controle, onde o produtor pode monitorar as métricas climáticas, verificar o estado físico dos equipamentos e gerenciar a operação remotamente. Este trabalho visa demonstrar a viabilidade da aplicação de tecnologias de Internet das Coisas (IoT) no agronegócio de pequeno porte, promovendo um manejo eficiente, reduzindo perdas produtivas e aumentando a eficiência operacional no campo.

Palavras-chave: sistema embarcado. sistema web. manejo térmico. suinocultura. automação.

ABSTRACT

Swine production plays a fundamental role in the Brazilian economy, being responsible for employment and income generation. In Pernambuco, the scenario is no different: although the state has a diversified economy and the sector is decentralized across its regions, various initiatives seek to boost this production, which is predominantly composed of small and medium-sized producers. The production cycle requires strict control over animal welfare to ensure the health and high performance of the herd. In this context, thermal comfort stands out as a critical factor for the proper development of the swine. Aiming to automate the climate control of the facilities, the present work proposes the development of a thermal management system that integrates an embedded module with a web platform. The hardware operates locally at the farm, using temperature sensors to autonomously activate or deactivate actuators. Simultaneously, the web application functions as a control panel, where the producer can monitor climate metrics, verify the physical status of the equipment, and manage operations remotely. This work aims to demonstrate the feasibility of applying Internet of Things (IoT) technologies in small-scale agribusinesses, promoting efficient management, reducing production losses, and increasing operational efficiency in the field.

Keywords: embedded system. web system. thermal management. swine production. automation.

LISTA DE FIGURAS

Figura 1 - Diagrama Entidade - Relacionamento.....	22
Figura 2 - Design da Interface em Fases Iniciais.....	23
Figura 3 - Cobertura de Testes Unitários com Jest.....	25
Figura 4 - MQTT Explorer conectado ao Eclipse Mosquitto.....	26
Figura 5 - Circuito de hardware no simulador Wokwi.....	29
Figura 6 - Protótipo de Hardware em Fases Iniciais.....	30
Figura 7 - Log do SE na plataforma Arduino IDE.....	31
Figura 8 - Diagrama de Casos de Uso do SW.....	36
Figura 9 - Diagrama de Blocos do SE.....	39
Figura 10 - Protótipo de hardware estágio final.....	45
Figura 11 - Tela Inicial.....	54
Figura 12 - Tela de Cadastro.....	54
Figura 13 - Tela de Recuperação de Senha.....	55
Figura 14 - Tela de Início do Sistema.....	55
Figura 15 - Tela de Início do Sistema (Continuação).....	56
Figura 16 - Tela de Configuração de Limiares.....	56
Figura 17 - Tela de Controle dos Atuadores - Modo Automático.....	57
Figura 18 - Tela de Controle dos Atuadores - Modo Manual.....	57
Figura 19 - Tela de Histórico de Notificações.....	58
Figura 20 - Tela de Informações do Perfil.....	58
Figura 21 - Tela de Informações do Perfil (Continuação).....	59
Figura 22 - Tela do Chat do Telegram com Notificações.....	60
Figura 23 - Notificação de Recuperação de Senha.....	60
Figura 24 - DER em Crow's Foot.....	61

LISTA DE TABELAS

Tabela 1 - Requisitos Funcionais do SW.....	32
Tabela 2 - Requisitos Não Funcionais do SW.....	34
Tabela 3 - Requisitos Funcionais do SE.....	34
Tabela 4 - Componentes de Hardware do SE.....	41
Tabela 5 - Cenários de teste de validação da comunicação MQTT.....	43
Tabela 6 - Cenários de Teste no Protótipo.....	44

SUMÁRIO

1 INTRODUÇÃO.....	10
2 OBJETIVOS.....	12
2.1 Objetivo Geral.....	12
2.2 Objetivos Específicos.....	12
2.3 Sistema de Automação do Manejo Térmico (SAMT).....	12
3 FUNDAMENTAÇÃO TEÓRICA.....	14
3.1 Sistemas Embarcados e Internet das Coisas (IoT).....	14
3.2 Microcontrolador: ESP32.....	14
3.3 Protocolo de Comunicação MQTT.....	15
3.4 Arquitetura de API REST.....	16
3.5 Sensores e Componentes de Verificação Física.....	17
3.6 Banco de Dados Relacional.....	18
4 TRABALHOS RELACIONADOS.....	19
5 METODOLOGIA.....	21
5.1 Sistema Web.....	23
5.1.1 Backend.....	23
5.1.2 Frontend.....	26
5.2 Sistema Embarcado.....	27
6 DOCUMENTAÇÃO.....	31
6.1 Requisitos Funcionais e Não Funcionais.....	31
6.1.1 Requisitos Funcionais do Sistema Web.....	31
6.1.2 Requisitos Não Funcionais do Sistema Web.....	32
6.1.3 Requisitos Funcionais do Sistema Embarcado.....	33
6.2 Diagrama de Casos de Uso do Sistema Web.....	35
6.2.1 Realizar Autenticação e Gerenciar Conta.....	35
6.2.2 Monitorar Granja.....	36
6.2.3 Controlar Atuadores Manualmente.....	36
6.2.4 Configurar Limiares de Temperatura.....	37
6.2.5 Consultar Histórico de Eventos.....	37
6.3 Diagrama de Blocos.....	38
6.3.1 Entradas.....	39
6.3.2 Saídas.....	39
6.3.3 Componentes de Hardware do Sistema Embarcado.....	40
7 RESULTADOS E DISCUSSÃO.....	41
7.1 Validação da API.....	41
7.2 Validação da Comunicação MQTT.....	42
7.3 Validação do Sistema Embarcado.....	43
7.4 Testes Integrados.....	44
7.5 Discussão dos Resultados.....	46

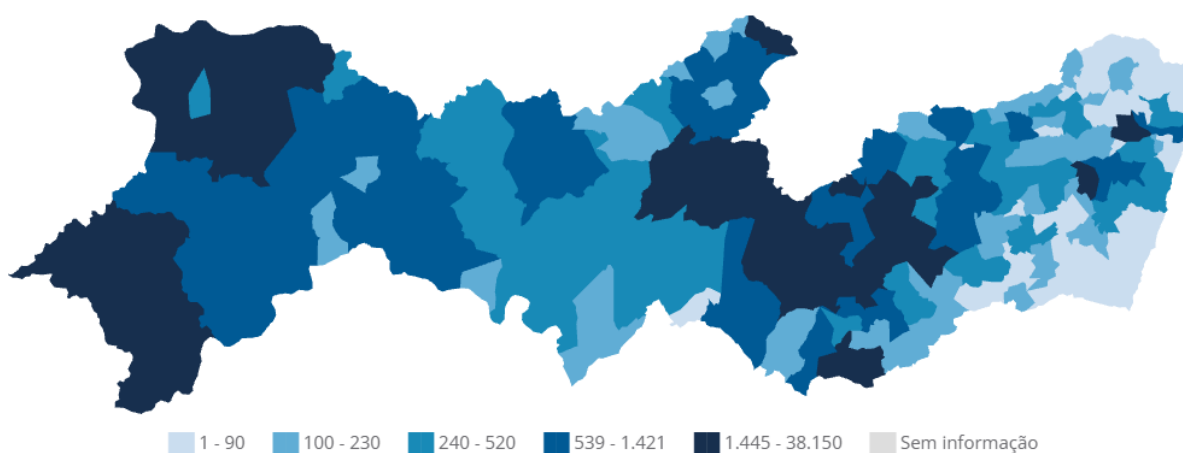
7.6 Limitações do Estudo.....	48
8 CONSIDERAÇÕES.....	50
REFERÊNCIAS.....	51
Apêndice A - Telas da Interface do Sistema Web.....	53
Apêndice B - Telas de Notificações.....	59
Apêndice C - Diagrama entidade relacionamento em notação crow's foot.....	60

1 INTRODUÇÃO

O efetivo suíno no Brasil em 2024 bateu a marca de aproximadamente 49,3 milhões de cabeças segundo a Pesquisa da Pecuária Municipal (PPM) realizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE). A suinocultura tem forte impacto na economia e geração de empregos no país, somente em 2024 a produção de carne suína atingiu o valor bruto de 59 milhões de reais (ABCS, 2025) e sete empregos indiretos são gerados a partir de um emprego formalizado (ABCS, 2024).

Em Pernambuco, a suinocultura também desempenha um papel importante principalmente para pequenos e médios produtores. A criação de suínos é descentralizada evidenciando a diversidade econômica do estado (ADEPE, 2024), porém cidades da região do Agreste têm forte participação no setor. A cidade de São Bento do Una se destaca com o maior rebanho com aproximadamente 38 mil matrizes (IBGE, 2024).

Mapa 1 - Rebanho de Suíno (cabeças) em Pernambuco



Fonte: PPM - Pesquisa da Pecuária Municipal (2024).

A região apresenta um grande potencial de desenvolvimento na criação de suínos recebendo investimentos como o projeto “Fortalecimento da Suinocultura Sustentável: inovação e valor na produção do Agreste Meridional” desenvolvido pelo Sebrae/PE e a Agência de Desenvolvimento Econômico de Pernambuco (ADEPE)

em 2025. Um dos objetivos do projeto foi implantar biodigestores nas propriedades participantes, visando a redução de até 15% dos custos de produção (FOLHA DE PERNAMBUCO, 2025). Esses equipamentos geram biogás a partir dos dejetos produzidos pelos suínos, o que colabora para o tratamento correto desses resíduos, e o gás pode ser utilizado como fonte de energia para outras atividades.

Através de projetos como esse é possível ver como a tecnologia e diferentes ferramentas podem ser inseridas em diversas etapas da cadeia de produção, oferecendo melhorias técnicas e sustentabilidade. No âmbito dos Objetivos de Desenvolvimento Sustentável (ODS) das Nações Unidas, diversas ações procuraram gerar práticas voltadas para a sustentabilidade, sendo o protocolo *SusPigSys* uma delas: desenvolvido na Europa para definir parâmetros sustentáveis que deveriam ser adotados na suinocultura do continente (ABCS, 2025).

O protocolo definiu 19 indicadores divididos em quatro áreas: sanidade e bem-estar animal, economia, meio-ambiente e social. O presente trabalho foca no indicador Conforto (térmico e físico, no descanso ou movimentação) enquadrado na primeira área. O conforto térmico é um dos pontos mais relevantes para o desempenho dos suínos, sem a correta ambientação térmica o estresse causado pode afetar a saúde e o desenvolvimento desses animais. Eles podem registrar alterações de comportamento, redução de peso na fase de crescimento e até aumento do número de natimortos (Beker *et al*, 2020).

2 OBJETIVOS

2.1 Objetivo Geral

Este trabalho tem como principal objetivo desenvolver um Sistema de Automação do Manejo Térmico (SAMT) de granjas de suínos que apoie o controle da temperatura nos ambientes onde os animais estão inseridos propiciando conforto térmico. O sistema é composto pela integração de um Sistema Embarcado (SE) e um Sistema Web (SW).

2.2 Objetivos Específicos

Os objetivos específicos deste trabalho estão listados a seguir:

- a) projetar o Sistema Embarcado (SE) integrando sensores de temperatura e componentes de acionamento para os atuadores de climatização;
- b) desenvolver o software do microcontrolador responsável pela leitura de dados e automação local do manejo térmico;
- c) construir o Sistema Web (SW) para atuar como interface gráfica centralizada de monitoramento em tempo real;
- d) integrar os sistemas (SE e SW) para viabilizar o controle remoto dos equipamentos e a parametrização dos limiares de temperatura;
- e) implementar sensores de verificação para monitorar o status operacional dos atuadores e garantir a integridade do sistema;

2.3 Sistema de Automação do Manejo Térmico (SAMT)

O desenvolvimento deste Sistema de Automação, unindo as capacidades do SE e do SW, visa proporcionar melhorias significativas no manejo térmico durante o ciclo produtivo. A aplicação dessas tecnologias garante o monitoramento e o controle térmico contínuo da granja, promovendo o bem-estar animal e impactando positivamente a saúde e o desempenho do rebanho.

O *software* desta solução atua como uma interface gráfica e painel de controle centralizado. Sua finalidade é permitir ao produtor o monitoramento em tempo real de métricas da granja, como temperatura e status operacional dos atuadores, além de possibilitar o controle remoto do Sistema Embarcado (SE), permitindo o acionamento direto de atuadores e a parametrização dos limiares de temperatura.

Por sua vez, o SE tem como finalidade automatizar o manejo térmico na granja por meio de seu *hardware* e *software*. Ele é o coração da solução: fará a leitura de temperatura através de sensores e realizará os comandos de ativação/desativação dos atuadores quando necessário. Para garantir a integridade do sistema, teremos sensores de verificação que monitoram o status operacional dos atuadores. Neste contexto, definem-se como atuadores os equipamentos de climatização da granja, especificamente os ventiladores e as lâmpadas de aquecimento.

3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os fundamentos teóricos e técnicos que embasaram as principais decisões arquiteturais do Sistema de Automação do Manejo Térmico (SAMT). Para cada tecnologia adotada são discutidas as alternativas disponíveis no mercado e os critérios técnicos que motivaram a escolha final, contemplando o microcontrolador, o protocolo de comunicação entre o Sistema Embarcado (SE) e o Sistema Web (SW), o estilo arquitetural da API, os sensores empregados e o modelo de persistência de dados.

3.1 Sistemas Embarcados e Internet das Coisas (IoT)

Um sistema embarcado é definido como um sistema computacional dedicado, projetado para executar uma função específica dentro de um sistema maior, geralmente sob restrições de custo, consumo energético e espaço físico. Quando esses sistemas são conectados à internet e capazes de coletar, processar e transmitir dados de forma autônoma, caracterizam a aplicação do paradigma de Internet das Coisas (IoT), no qual objetos físicos passam a interagir entre si e com plataformas de software por meio de redes de comunicação. No contexto do SAMT, a adoção de um sistema embarcado conectado permite que o monitoramento e o acionamento dos equipamentos de climatização ocorram de forma local e em tempo real, mesmo quando a conectividade com a internet é instável, ao mesmo tempo em que viabiliza a integração com uma plataforma web para o monitoramento remoto.

3.2 Microcontrolador: ESP32

A escolha do microcontrolador é uma decisão central no projeto de um sistema embarcado, pois define os limites de processamento, conectividade e quantidade de periféricos que podem ser controlados simultaneamente. Dentre as opções amplamente utilizadas em projetos de baixo custo, destacam-se os microcontroladores da família Espressif, ESP8266 e ESP32. O ESP8266 possui um processador de núcleo único, operando em até 160 MHz, e disponibiliza um número reduzido de pinos de entrada e saída (GPIO), o que restringe a quantidade de

sensores e atuadores que podem ser conectados diretamente à placa. O ESP32, por sua vez, é equipado com processador de arquitetura dual-core, de até 240 MHz, maior quantidade de memória RAM, conversores analógico-digitais adicionais e um número significativamente maior de pinos GPIO, além de contar com conectividade Wi-Fi e Bluetooth nativa (ESPRESSIF SYSTEMS, 2024).

Para o SAMT, essas características foram decisivas: o sistema necessita gerenciar simultaneamente a leitura periódica do sensor de temperatura e umidade, a leitura dos sensores de verificação física (LDR e Hall), o acionamento de dois módulos relé e a manutenção estável da conexão Wi-Fi e da sessão com o broker MQTT. A arquitetura dual-core do ESP32 permite distribuir essas tarefas entre os núcleos do processador sem comprometer o tempo de resposta do sistema, algo que seria mais suscetível a instabilidades em um microcontrolador de núcleo único como o ESP8266, especialmente em cenários de expansão futura do projeto para múltiplos dispositivos e sensores adicionais, como o sensor de umidade.

3.3 Protocolo de Comunicação MQTT

A comunicação entre o Sistema Embarcado e o Sistema Web exige um protocolo de mensageria que seja leve, tolerante a instabilidades de rede e adequado às restrições de processamento e energia de dispositivos embarcados. Dentre as alternativas consideradas estavam a comunicação via requisições HTTP recorrentes (*polling*), o uso de WebSocket e o protocolo MQTT (Message Queuing Telemetry Transport). O modelo de *polling* via HTTP foi descartado por gerar sobrecarga desnecessária de requisições, já que o cliente precisaria consultar o servidor em intervalos fixos mesmo quando não houvesse mudança de estado, aumentando o consumo de energia e de dados do dispositivo embarcado. O WebSocket, embora ofereça comunicação bidirecional persistente, não possui nativamente mecanismos de qualidade de serviço (QoS) e retenção de mensagens, características relevantes para um sistema de controle que precisa garantir a entrega de comandos.

O MQTT foi adotado por seguir o padrão de comunicação *publish/subscribe*, que desacopla o produtor da informação (o SE, ao publicar leituras de temperatura e status dos atuadores) do consumidor (o SW, ao assinar os tópicos

correspondentes), permitindo que múltiplos dispositivos publiquem e vários clientes se inscrevam nos mesmos tópicos sem uma conexão direta entre eles. O protocolo é reconhecido como padrão OASIS voltado a cenários de rede restrita e conexão intermitente, com cabeçalhos reduzidos e suporte a diferentes níveis de qualidade de serviço (MQTT.ORG, [s. d.]). Essas características se mostraram diretamente aplicáveis ao SAMT: a flag de retenção (*retain*) foi utilizada para garantir que o SE receba a última configuração de limiares de temperatura mesmo que esteja *offline* no momento da alteração, e o modelo de tópicos permitiu isolar as mensagens de telemetria (`granja/suinos/status/disp_{id}`) das mensagens de configuração e comando (`granja/suinos/config/disp_{id}`), mantendo a comunicação organizada e de baixa latência.

3.4 Arquitetura de API REST

A definição da arquitetura de comunicação entre o *frontend* e o *backend* do Sistema Web foi motivada pela necessidade de suportar múltiplos tipos de cliente de forma desacoplada, permitindo que cada camada evoluísse de maneira independente. Entre as alternativas arquiteturais avaliadas estavam o padrão SOAP, o framework gRPC (Google Remote Procedure Call) e o estilo arquitetural REST (Representational State Transfer).

O SOAP foi descartado por sua maior complexidade de implementação, pela verbosidade do formato XML utilizado em suas mensagens e pelo *overhead* de processamento. O gRPC, por sua vez, é um *framework* de chamada de procedimento remoto de alto desempenho que utiliza HTTP/2 e Protocol Buffers como formato de serialização, exigindo a definição prévia de contratos de serviço em arquivos de esquema (.proto) e a geração de código cliente/servidor a partir deles (gRPC, [s. d.]). Embora ofereça vantagens de desempenho relevantes para comunicação entre serviços internos, o gRPC não é nativamente suportado por navegadores web, exigindo uma camada adicional de tradução (gRPC-Web) e um proxy intermediário para viabilizar chamadas a partir do frontend. Essa exigência adicionaria complexidade de infraestrutura incompatível com o escopo do projeto, cujo cliente principal é uma aplicação React executada diretamente no navegador.

O estilo REST, descrito originalmente por Fielding (2000), propõe a comunicação stateless entre cliente e servidor por meio de operações padronizadas (GET, POST, PUT, DELETE) sobre recursos identificados por URIs, o que se alinhou diretamente aos requisitos do projeto: cada entidade do domínio (produtor, dispositivo, evento, notificação) pôde ser mapeada como um recurso independente, manipulado por meio de operações CRUD (Create, Read, Update, Delete) padronizadas. A característica stateless do REST também contribuiu para o desacoplamento entre o backend e o frontend, pois cada requisição contém toda a informação necessária para seu processamento (incluindo o token JWT de autenticação), sem depender de estado mantido em sessão no servidor. Por fim, a maturidade e a ampla adoção do padrão REST no mercado favoreceram a escolha, uma vez que estão disponíveis ferramentas consolidadas de teste e documentação, como o Postman, utilizado neste trabalho para a validação dos endpoints da API.

3.5 Sensores e Componentes de Verificação Física

A seleção dos sensores do Sistema Embarcado também decorreu de uma análise comparativa de alternativas. Para a leitura ambiental, considerou-se o uso do sensor DHT11, mais simples e de menor custo, e do sensor DHT22 (também comercializado como AM2302), efetivamente adotado no projeto. O DHT22 oferece maior faixa de operação e maior precisão de leitura de temperatura em comparação ao DHT11, características relevantes para um sistema cuja lógica de acionamento automático dos atuadores depende diretamente da exatidão da métrica coletada (AOSONG ELECTRONICS, [s. d.]). Embora o DHT22 também disponibilize a leitura de umidade relativa do ar, essa métrica não foi incorporada à lógica de acionamento do ventilador nesta versão do sistema, uma vez que o escopo do trabalho priorizou a validação do ciclo completo de automação a partir de um único critério de controle (a temperatura).

Além do sensor ambiental, o projeto incorporou os sensores LDR e Hall não para o controle climático em si, mas como uma camada de verificação física do estado dos atuadores. Essa decisão se fundamenta no princípio de que, em sistemas de automação que acionam cargas físicas por meio de relés, o comando lógico enviado ao atuador não garante, por si só, que o equipamento tenha

efetivamente mudado de estado, uma vez que falhas mecânicas, elétricas ou de fiação podem ocorrer sem que o microcontrolador tenha meios de detectá-las. O sensor LDR, posicionado próximo à lâmpada de aquecimento, permite inferir seu funcionamento pela variação de luminosidade captada, enquanto o sensor Hall, posicionado próximo à ventoinha, permite inferir sua rotação pela detecção de pulsos magnéticos. Essa redundância sensorial confere ao sistema maior confiabilidade operacional, viabilizando a geração de alertas de falha de hardware ao produtor.

3.6 Banco de Dados Relacional

A escolha de um Sistema Gerenciador de Banco de Dados (SGBD) relacional, em detrimento de alternativas não relacionais (NoSQL), foi motivada pela natureza estruturada e fortemente relacionada dos dados manipulados pelo SAMT, conforme representado no Diagrama Entidade-Relacionamento (Figura 1). Entidades como Produtor, Dispositivo, Evento, Atuador e Notificação possuem relacionamentos bem definidos entre si, com cardinalidades específicas que se beneficiam de um esquema rígido e de mecanismos de integridade referencial, nativamente suportados por bancos relacionais. Adicionalmente, a exigência de garantir a consistência dos registros de eventos de climatização, mesmo diante de operações concorrentes, motivou a adoção de um SGBD compatível com propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), tal como o MySQL, permitindo o uso de transações para assegurar que o registro de um evento e o disparo de sua respectiva notificação ocorram de forma íntegra.

4 TRABALHOS RELACIONADOS

Dada a relevância da ambiência na suinocultura, em especial o controle térmico das instalações para a garantia do bem-estar animal e a otimização dos índices produtivos, somada ao valor estratégico que a tecnologia agrega ao setor agropecuário, diferentes pesquisas têm proposto soluções automatizadas para essa demanda.

Silva *et al.* (2022) desenvolveram um sistema de monitoramento e controle de baixo custo voltado para instalações de suínos na fase de maternidade. Os testes foram conduzidos em ambiente real no setor de Suinocultura da Fazenda Lageado, pertencente à Universidade Estadual Paulista (UNESP), possibilitando a validação prática da solução proposta.

Em termos estruturais, os autores construíram um sistema de hardware baseado no microcontrolador Arduino Leonardo integrado ao sensor de temperatura e umidade DHT11. A lógica programada atua no acionamento automático dos ventiladores da maternidade sempre que a temperatura ambiente atinge o limiar estipulado na programação.

No protótipo de Silva *et al.* (2022), as alterações nos limites de temperatura programados são realizadas localmente por meio de botões pulsadores e um display LCD fixados no próprio *hardware*. A conclusão do estudo destaca que a aplicação de sistemas de automação acessíveis é viável e gera impactos práticos positivos para o ciclo produtivo, modernizando o manejo diário.

O presente trabalho diferencia-se da abordagem de Silva *et al.* (2022) pela conectividade e persistência de dados. Enquanto a tecnologia correlata limita-se ao controle estritamente local, esta pesquisa propõe o Sistema de Automação do Manejo Térmico (SAMT), que integra o Sistema Embarcado (SE) a um Sistema Web (SW). Essa arquitetura permite o monitoramento em tempo real e parametrização remota via painel centralizado, além de utilizar um banco de dados dedicado para registrar o histórico de ações dos atuadores, garantindo maior rastreabilidade ao manejo.

Em outra abordagem voltada para as necessidades de pequenas propriedades rurais, Borges *et al.* (2018) propuseram o desenvolvimento de um

sistema automatizado de baixo custo para o controle e o monitoramento do ambiente térmico em instalações suínolas. O funcionamento lógico da pesquisa baseou-se no cálculo da entalpia (correlacionando os níveis de temperatura e umidade) para acionar equipamentos apenas quando o ambiente se encontrasse fora das faixas ideais de conforto térmico.

Para a construção do *hardware*, os autores utilizaram a plataforma Arduino integrada a sensores, relés e atuadores, mais especificamente um ventilador e um nebulizador. O sistema foi programado para realizar leituras a cada 10 segundos e armazenar a média dessas medições a cada 10 minutos em um módulo de cartão de memória SD (*Secure Digital*) com o auxílio de um módulo de relógio interno (RTC DS3231).

A solução de Borges *et al.* (2018) obteve êxito ao estabilizar o clima do ambiente após cerca de duas horas e meia de operação em condições críticas, promovendo a economia de energia e água por acionar os equipamentos apenas quando estritamente necessário. Destaca-se também a autonomia estrutural do projeto: ao contrário de outros sistemas que exigem um computador dedicado conectado via porta serial para registrar os dados, a proposta dos autores é independente e pode ser alimentada por um simples carregador de celular de 5V.

Apesar da eficiência no controle local, o armazenamento de dados via cartão SD em Borges *et al.* (2018) exige a extração física da memória para análise manual dos dados em planilhas. O SAMT supera essa limitação logística ao ligar o SE ao SW, eliminando mídias removíveis. Os registros são salvos diretamente em banco de dados, oferecendo ao produtor uma interface gráfica robusta para visualização de métricas em tempo real e parametrização remota, garantindo uma gestão centralizada e imediata da granja.

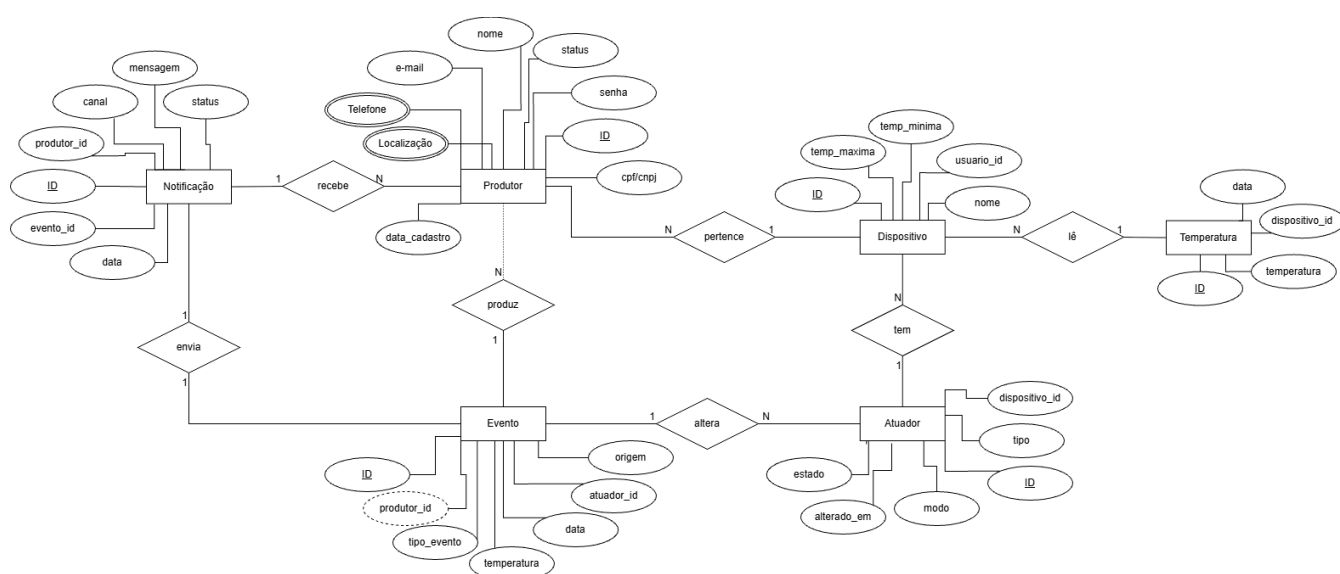
Os trabalhos analisados evidenciam a viabilidade da aplicação de tecnologias embarcadas na automação do manejo térmico da suinocultura. Entretanto, observa-se que as soluções existentes concentram-se predominantemente no controle local ou em mecanismos limitados de armazenamento e consulta dos dados. Nesse contexto, o Sistema de Automação do Manejo Térmico (SAMT) proposto neste trabalho busca agregar recursos de conectividade, monitoramento remoto, persistência de dados e notificação em tempo real, ampliando as possibilidades de gestão e acompanhamento do ambiente produtivo.

5 METODOLOGIA

Este trabalho caracteriza-se como um projeto tecnológico de natureza aplicada, com abordagem prática baseada no desenvolvimento e validação de um protótipo funcional. O presente projeto é composto por dois sistemas integrados: O Sistema Web (SW) e o Sistema Embarcado (SE). Sendo assim, o processo de desenvolvimento dele foi separado em duas camadas permitindo que as particularidades de cada frente fossem realizadas de forma especializada. O início do processo de desenvolvimento se deu pelo levantamento de requisitos e definição de tecnologias e ferramentas a serem utilizadas. Nesta fase, foram definidos os requisitos funcionais e não funcionais de cada uma das camadas, bem como os equipamentos para a construção do protótipo físico do Sistema Embarcado.

Após finalizada a fase de levantamento de requisitos e definição de tecnologias, o desenvolvimento prosseguiu para a modelagem de dados e o design de interface do Sistema Web. Diante da característica estrutural dos dados a serem coletados e produzidos pelos sistemas, optou-se pelo uso de um banco de dados relacional. Assim, foi construído o Diagrama Entidade-Relacionamento (DER) (Figura 1) para representar as entidades, atributos e relações dos dados no sistema.

Figura 1 - Diagrama Entidade - Relacionamento

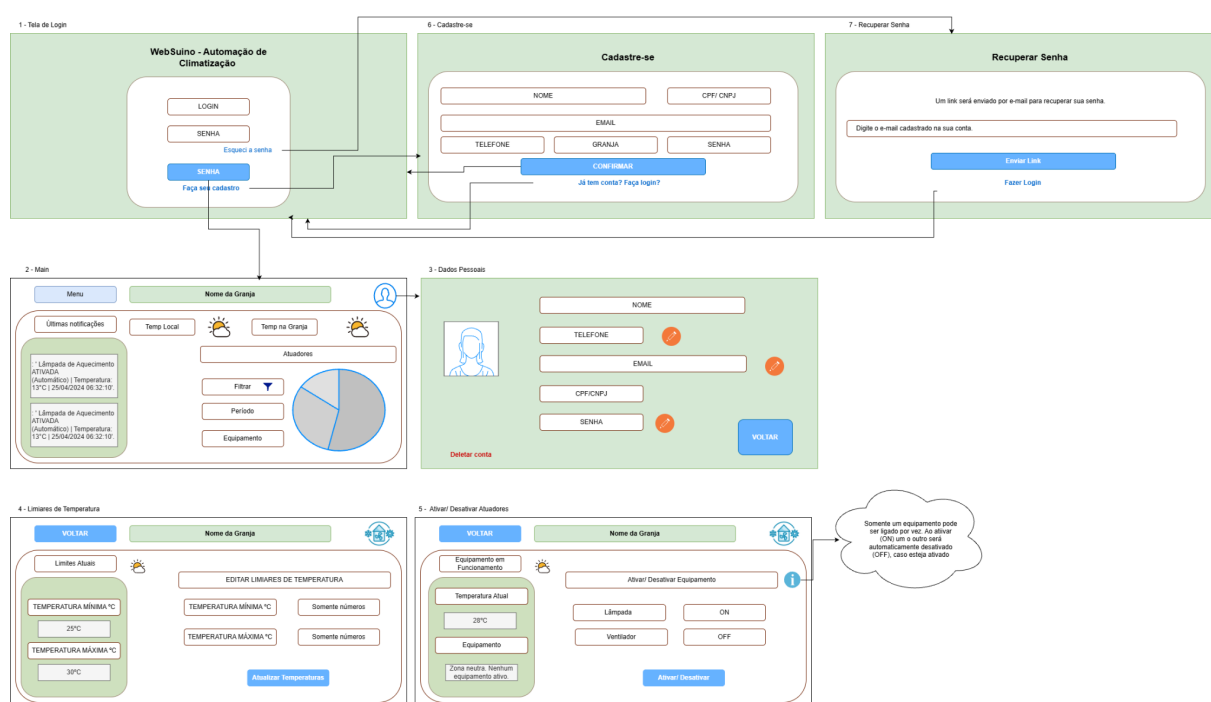


Fonte: Autora (2026).

A elaboração do DER, em paralelo, ao design da interface proporcionou clareza ao processo de desenvolvimento. Essa abordagem viabilizou a definição precisa dos dados essenciais para o funcionamento do sistema, além de mitigar possíveis redundâncias, resultando em uma modelagem consistente e normalizada.

A Figura 2 mostra a versão inicial da interface web do sistema. À medida que o desenvolvimento foi evoluindo novas telas foram adicionadas a ela, e a interface final e completa pode ser visualizada no Apêndice A.

Figura 2 - Design da Interface em Fases Iniciais



Fonte: Autora (2026).

Concluídas as duas fases anteriores, iniciou-se a fase de desenvolvimento de cada camada do sistema de forma isolada, ou seja, sem interação ou compartilhamento de dados entre elas até a última etapa, momento em que ocorreu a integração do sistema de ponta a ponta. O isolamento das camadas permitiu que cada uma delas fosse testada e aperfeiçoada sem interferência de fatores externos que pudessem comprometer seu funcionamento.

5.1 Sistema Web

O sistema web¹ foi desenvolvido sob uma Arquitetura MVC (*Model, View, Controller*) desacoplada, onde o backend e o frontend são construídos separadamente, e sua comunicação ocorre via requisições HTTP (*Hypertext Transfer Protocol*) trocando dados no formato JSON (*JavaScript Object Notation*).

O JavaScript foi a linguagem de programação escolhida para o desenvolvimento de todo o sistema web. Os principais motivos para a escolha foram a sua versatilidade e a possibilidade de usar a mesma linguagem em todas as camadas do sistema, além da maturidade e confiabilidade da tecnologia no mercado.

5.1.1 Backend

O *backend* foi construído através do *framework* Node.js² e baseado em *Application Programming Interface (API) REST*. A escolha dessa arquitetura foi motivada pelas suas características *stateless* e, principalmente, o desacoplamento cliente/servidor que contribuiu para a separação estrita das responsabilidades entre o *backend* e o *frontend*. Para o gerenciamento das rotas da API e a conectividade entre as camadas, utilizou-se o *framework* Express.js³.

A lógica de negócio do *backend*, os *controllers*, foi validada com o *framework* Jest, que é amplamente consolidado no mercado e oferece suporte a diferentes *frameworks* de desenvolvimento, como o Node.js. Os testes unitários implementados com o Jest garantiram a validação das lógicas de negócio e as travas de segurança através de *mocks* do banco de dados.

A fim de aferir a eficácia da cobertura dos testes, aplicou-se a técnica de Teste de Mutação em parte dos *controllers* da API. A inserção de falhas lógicas no

¹ Repositório contendo a implementação completa do Sistema de Automação do Manejo Térmico (SAMT). Disponível em: <https://github.com/ElianeCordeiro/sistema-manejo-termico>. Acesso em: 14 jun. 2026

² Node.js. Documentação oficial da plataforma de execução JavaScript. Disponível em: <https://nodejs.org/>. Acesso em: 14 jun. 2026

³ Express.js. Documentação oficial do framework para desenvolvimento de aplicações web com Node.js. Disponível em: <https://expressjs.com/pt-br/>. Acesso em: 14 jun. 2026

código fonte resultou na rejeição imediata dos cenários de teste correspondentes, comprovando que a suíte de testes com o Jest é eficaz e robusta para atestar a qualidade do código.

Figura 3 - Cobertura de Testes Unitários com Jest

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	63.41	70.3	36.2	63.71	
config	100	100	100	100	
db.js	100	100	100	100	
controllers	92.47	86.91	100	92.39	
AtuadorController.js	96.87	82.69	100	96.77	95-96
AuthController.js	100	93.33	100	100	21
ConfiguracaoController.js	80.64	85.71	100	80.64	11,22-23,34,57-58
DashboardController.js	89.47	95.45	100	89.09	22-23,61-62,115-116
LeituraController.js	97.36	88.88	100	97.36	46
NotificacaoController.js	100	100	100	100	
PasswordController.js	82.92	80	100	82.92	12,61,67-68,77,96-97
ProdutorController.js	93.67	84.21	100	93.67	26-27,60-64
models	8.39	0	0	8.69	
AtuadorModel.js	10	0	0	10	5-40
ConfiguracaoModel.js	25	100	0	25	5-15
DashboardModel.js	7.4	0	0	8.69	5-101
LeituraModel.js	7.4	0	0	7.69	5-59
NotificacaoModel.js	20	0	0	20	5-16
ProdutorModel.js	3.92	0	0	3.92	5-103
services	17.24	3.57	0	17.24	
MqttService.js	13.95	0	0	13.95	10-92
TelegramService.js	26.66	12.5	0	26.66	8-26
Test Suites: 8 passed, 8 total Tests: 72 passed, 72 total Snapshots: 0 total Time: 10.824 s					

Fonte: Autora (2026).

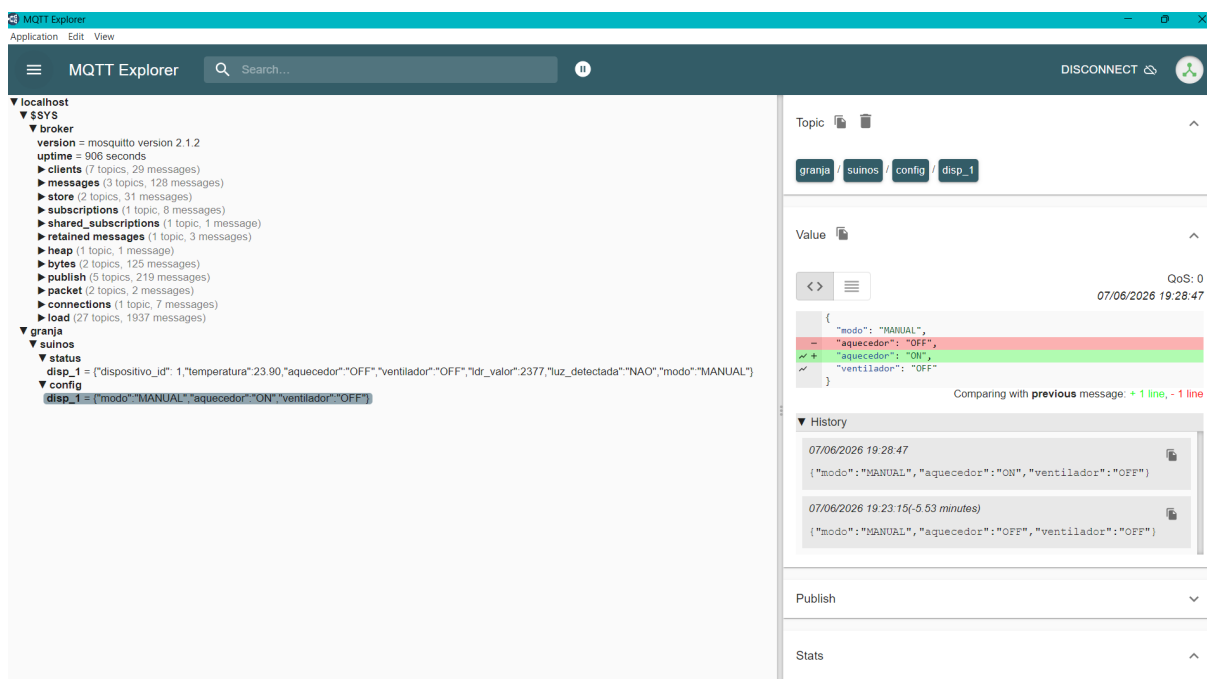
O Postman foi escolhido para validar os *endpoints* da API REST, simulando o comportamento do cliente (*frontend*) antes da integração das camadas. Na camada de segurança da API, foi implementado o *JSON Web Token* (JWT), para proteger as rotas da API e manter a sessão do usuário, além da biblioteca *bcrypt*, empregada para a criptografia das senhas no banco de dados.

A comunicação do *backend* com o sistema embarcado foi implementada para utilizar o protocolo MQTT (*Message Queuing Telemetry Transport*). Trata-se de um protocolo leve e que oferece baixa latência, ideal para ecossistemas de Internet das Coisas (IoC). Durante a fase inicial de prototipagem com o simulador de circuitos

Wokwi, utilizou-se o *broker* público HiveMQ para intermediar a troca de mensagens e validar a lógica de integração.

Nos estágios finais de desenvolvimento, com a implantação do hardware físico, a infraestrutura de mensageria foi migrada para o serviço Eclipse Mosquitto. Diferentemente do ambiente público e gratuito utilizado nos testes, o Mosquitto permitiu a configuração de um *broker* privado com autenticação e isolamento de rede, garantindo a segurança dos dados e impedindo que agentes externos não autorizados acessassem ou manipulassem os tópicos de controle do sistema. Para suprir a ausência de uma interface gráfica nativa neste serviço, adotou-se o *software* cliente MQTT Explorer. Essa ferramenta atuou em conjunto com a interface de linha de comando (CLI) para possibilitar o monitoramento visual e a auditoria em tempo real do tráfego de dados.

Figura 4 - MQTT Explorer conectado ao Eclipse Mosquitto



Fonte: Autora (2026).

O sistema gerenciador de banco de dados (SGBD) escolhido para fazer parte do sistema foi o MySQL, em sua versão *open source*, MySQL *Community*. Trata-se de um SGBD escalável, amplamente utilizado em aplicações modernas e, além disso, é compatível com propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade). Essa característica permitiu garantir a integridade dos dados

através do uso de transações (BEGIN, COMMIT) ao longo do desenvolvimento das regras de negócio.

O envio de notificações em tempo real foi implementado através da integração com a API oficial do Telegram. A comunicação entre o *backend* e a plataforma de mensagens ocorre por meio de requisições HTTP gerenciadas por uma classe de serviço estática, mantendo essa função isolada dos controladores principais. Essa estrutura permite o disparo de alertas instantâneos para o dispositivo do produtor, utilizando o *Chat ID* armazenado no banco de dados, sempre que o estado dos atuadores sofre uma alteração ou quando a ESP32 emite um aviso de falha física.

A funcionalidade de recuperação de credenciais foi desenvolvida com base no protocolo *Simple Mail Transfer Protocol* (SMTP). O *backend* utiliza este protocolo para estabelecer a ligação com servidores de correio eletrónico e disparar mensagens automatizadas contendo os *links* para a redefinição de acesso. O uso do SMTP garante um fluxo seguro e autônomo para que o usuário redefina a sua senha, validando a requisição diretamente na caixa de entrada do e-mail antes de atualizar os dados na tabela correspondente do MySQL.

5.1.2 Frontend

A camada de visualização do sistema foi desenvolvida com React⁴ e Vite⁵. A biblioteca React é de código aberto e amplamente utilizada para construção de interfaces de usuário, oferece suporte a diversas bibliotecas e ferramentas. O Vite, por sua vez, otimiza a execução e compilação do código, oferecendo alta performance no desenvolvimento. A combinação das duas ferramentas viabilizou um ambiente ágil e robusto.

Para a renderização do histórico de temperatura na interface *web*, utilizou-se a biblioteca Recharts⁶. Essa integração proporcionou uma visualização mais

⁴ React. Documentação oficial da biblioteca para construção de interfaces de usuário. Disponível em <https://react.dev/>. Acesso em: 14 jun. 2026.

⁵ Vite. Documentação oficial da ferramenta de build e desenvolvimento frontend. Disponível em <https://pt.vite.dev/guide/>. Acesso em: 14 jun. 2026

⁶ Recharts. Documentação oficial da biblioteca para visualização de dados em aplicações. Disponível em <https://recharts.github.io/en-US/guide/>. Acesso em 14 jun. 2026.

amigável dos dados do sistema, diversificando as formas de apresentação das métricas coletadas, como gráficos.

Durante o desenvolvimento da interface, a ferramenta de inteligência artificial Gemini, baseada em modelos de linguagem, foi utilizada como apoio em decisões de design visual, auxiliando na definição de paletas de cores e no posicionamento de elementos na tela. O uso da ferramenta se restringiu a sugestões estéticas e de layout. Ademais, todas as sugestões geradas foram avaliadas, ajustadas e incorporadas manualmente pela autora.

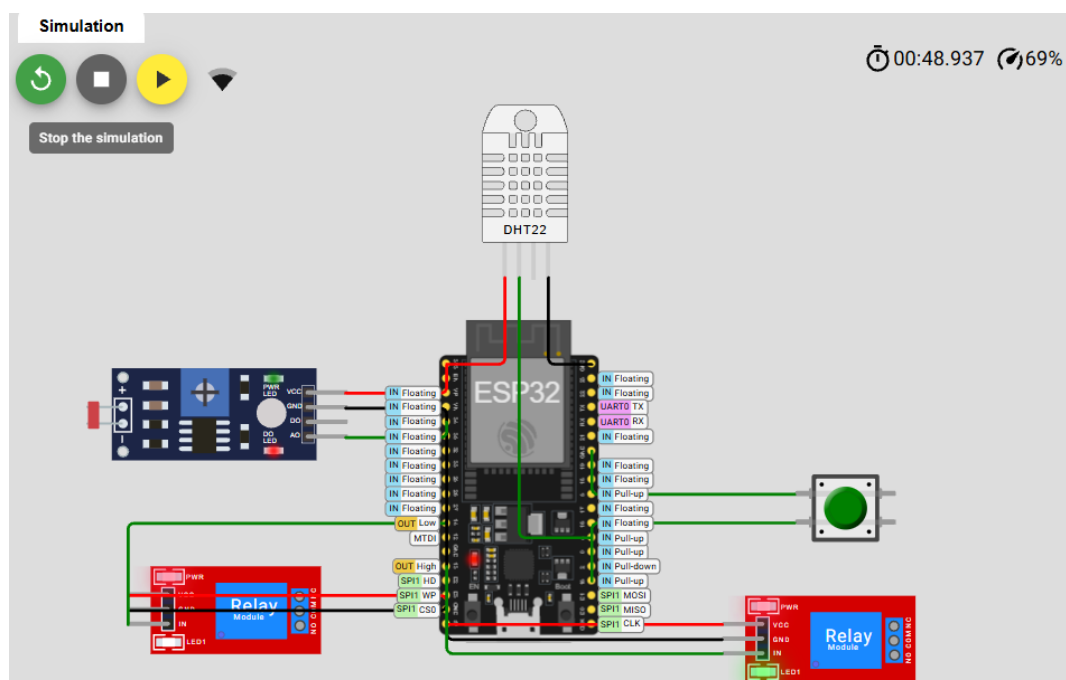
5.2 Sistema Embarcado

O sistema embarcado, dividido em *software* e *hardware*, teve o ciclo de vida do projeto separado em cinco fases. As fases 1 e 2, que compreendem, respectivamente, a especificação do produto e o particionamento do projeto em camadas de *software* e *hardware*, foram realizadas nos passos iniciais do desenvolvimento, conforme descrito anteriormente.

A terceira fase consistiu na simulação do sistema em um ambiente virtual e na construção do protótipo físico, que permitiu a emulação e a validação do código e dos componentes. A ferramenta escolhida para essa fase foi a Wokwi, um simulador de circuitos eletrônicos e microcontroladores. Por meio dela foi possível implementar a lógica do *firmware* e construir a interação dos componentes (microcontrolador, sensores e relé) nas simulações⁷.

⁷ Simulação do circuito e do código-fonte desenvolvidos para este projeto no Wokwi. Disponível em: <https://wokwi.com/projects/464582460726201345>. Acesso em: 14 jun. 2026.

Figura 5 - Circuito de *hardware* no simulador Wokwi

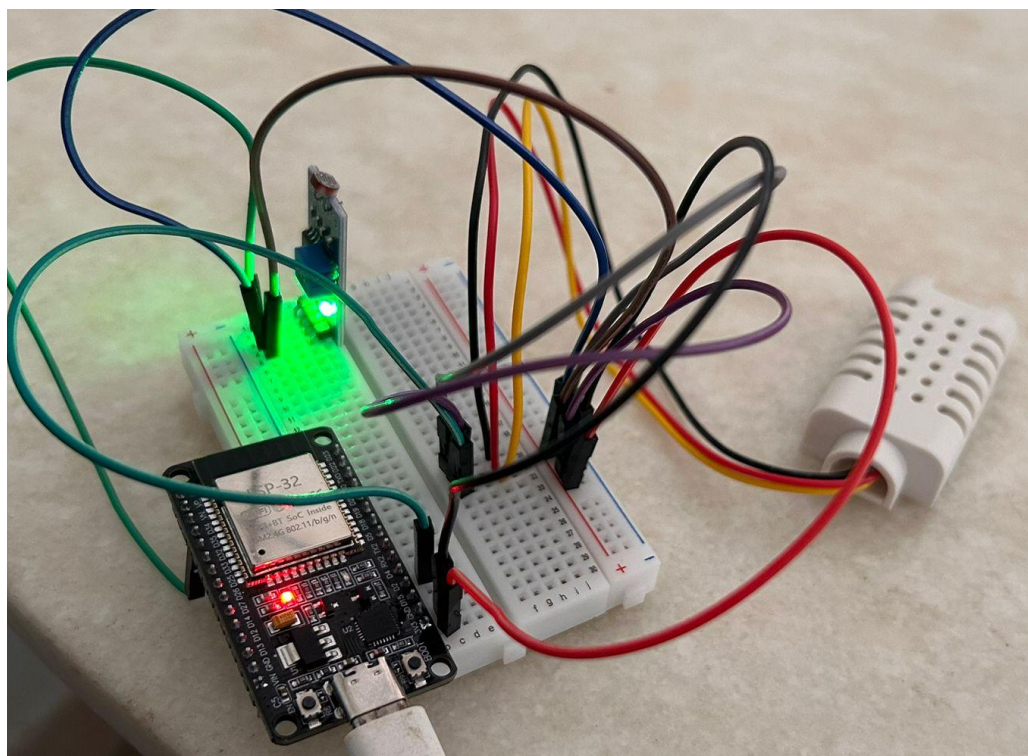


Fonte: Autora (2026).

Devido a limitações da ferramenta, o componente de sensor hall, responsável por checar o funcionamento do ventilador, foi simulado através do *button* e lógica de programação na plataforma. A simulação completa na plataforma do Wokwi⁷.

A construção do protótipo físico foi desdobrada em pequenas etapas devido à quantidade de componentes e à necessidade de checar o correto funcionamento de cada um deles individualmente. A Figura 4 ilustra uma das fases de construção do protótipo, na qual é possível visualizar, da esquerda para a direita, o microcontrolador ESP32, o sensor de luminosidade LDR e o sensor de temperatura e umidade DHT22 integrados a uma *protoboard* através de *jumpers*.

Figura 6 - Protótipo de Hardware em Fases Iniciais



Fonte: Autora (2026).

A portabilidade do código desenvolvido no simulador do Wokwi ao ESP32 ocorreu na quarta fase do desenvolvimento. Nesta etapa, foram realizados testes e adaptações para garantir que o comportamento em conjunto dos componentes estava funcionando corretamente. Na Figura 5, é apresentado o *log* do Sistema Embarcado (SE) confirmando a conexão ao *broker* MQTT, a publicação e o

recebimento de mensagens, a leitura da temperatura e o armazenamento de novos limiares térmicos na memória não volátil.

Figura 7 - Log do SE na plataforma Arduino IDE

```

le Edit Sketch Tools Help
ESP32 Dev Module

sketch_may26a.ino
64 void setup() {
65   Serial.begin(115200);
66
67   delay(3000);
68   Serial.println("\n\n--- INICIANDO SISTEMA ---");
69
70   pinMode(PINO_AQUECEDOR, OUTPUT);
71   pinMode(PINO_VENTILADOR, OUTPUT);
72   digitalWrite(PINO_AQUECEDOR, LOW);
73   digitalWrite(PINO_VENTILADOR, LOW);
74 }

Output Serial Monitor X
Not connected. Select a board and a port to connect automatically.
New Line

w1-r1 Conectado:
Tentando conexão MQTT...Conectado ao Broker MQTT!
Assinado no tópico de configuração.
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.70,"aquecedor":"ON","ventilador":"OFF"}
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}

Mensagem recebida no tópico: granja/suinos/config/disp_1
Payload: {"temp_minima":28,"temp_maxima":32}
Novos limiares salvos na RAM e na Flash
Mínima: 28.00 | Máxima: 32.00
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"ON","ventilador":"OFF"}

Mensagem recebida no tópico: granja/suinos/config/disp_1
Payload: {"temp_minima":26,"temp_maxima":30}
Novos limiares salvos na RAM e na Flash
Mínima: 26.00 | Máxima: 30.00
Publicando no tópico granja/suinos/status/disp_1 -> {"dispositivo_id": 1,"temperatura":27.80,"aquecedor":"OFF","ventilador":"OFF"}
  
```

Fonte: Autora (2026).

A linguagem de programação C/C++ foi a escolhida para a elaboração do *firmware* do SE e a plataforma Arduino IDE foi o ambiente de desenvolvimento na etapa de prototipação física do sistema.

Após constatado o sucesso na integração de todos os componentes, a fase final da construção do SE consistiu na conexão entre este e o sistema web. Esta etapa consolidou a comunicação de todo o sistema de climatização de ponta a ponta: a leitura da temperatura pelo sensor DHT22, a mudança de estado dos atuadores, o compartilhamento dos dados via *broker*, processamento dos dados no *backend* e a exibição na *interface* do usuário. Com esta fase, encerrou-se o ciclo de desenvolvimento de todas as camadas do projeto.

6 DOCUMENTAÇÃO

6.1 Requisitos Funcionais e Não Funcionais

Esta seção tem como objetivo estabelecer as especificações técnicas e operacionais que guiaram o desenvolvimento do Sistema de Automação do Manejo Térmico. A etapa de levantamento de requisitos é crucial para alinhar o escopo técnico aos objetivos do projeto, garantindo que a integração entre o *software* e *hardware* ocorra de maneira coesa.

6.1.1 Requisitos Funcionais do Sistema Web

A Tabela 1 apresenta os Requisitos Funcionais do Sistema Web, cujo objetivo é mapear todas as funcionalidades, ações e interações que a plataforma deve oferecer ao usuário, englobando desde a autenticação no sistema até o monitoramento das métricas climáticas e o controle remoto dos atuadores da granja.

Tabela 1 - Requisitos Funcionais do SW

ID	Nome	Descrição	Prioridade
RFSW01	Visualizar atualizações de temperatura	O sistema deve exibir as últimas leituras de temperatura.	Alta
RFSW02	Controle manual dos atuadores	O sistema deve permitir que o usuário autenticado ligue ou desligue manualmente a lâmpada de aquecimento ou o ventilador, sobrescrevendo o modo automático. Ao ligar um atuador, o outro deve ser desligado automaticamente (exclusão mútua).	Alta
RFSW03	Cadastro e autenticação	O sistema deve permitir cadastro de usuários e autenticação com e-mail e senha, utilizando JWT para controle de sessão.	Alta

RFSW04	Persistência de dados	Todos os eventos de climatização de ambos os atuadores devem ser armazenados em banco de dados relacional MySQL.	Alta
RFSW05	Enviar notificações	O sistema deve enviar notificações via Telegram ao produtor sempre que o estado de qualquer atuador mudar, indicando: equipamento afetado (lâmpada ou ventilador), ação realizada (ativado/desativado), origem do comando (automático ou manual), temperatura atual e data/hora.	Alta
RFSW06	Histórico de eventos	O sistema deve exibir lista paginada com o histórico de eventos, com filtro por período.	Alta
RFSW07	Configuração limiares de temperatura	O sistema deve permitir que o usuário configure os limiares de temperatura: temperatura mínima (abaixo da qual a lâmpada é ativada) e temperatura máxima (acima da qual o ventilador é ativado). Esses valores devem ser persistidos no banco e enviados ao SE via MQTT sempre que alterados.	Alta
RFSW08	Recuperação de senha	O sistema deve permitir que o usuário redefina sua senha via link enviado por e-mail.	Alta
RFSW09	Notificação de falha de atuador	Ao receber mensagem de falha do atuador, o backend deve registrar o evento no banco de dados e enviar imediatamente uma notificação ao produtor.	Alta

Fonte: Autora (2026).

6.1.2 Requisitos Não Funcionais do Sistema Web

A Tabela 2 apresenta os Requisitos Não Funcionais do Sistema Web, cujo objetivo é estabelecer as métricas e restrições de qualidade da aplicação, focando no desempenho da comunicação em tempo real, na segurança por meio de autenticação e criptografia de dados, além da adoção de padrões arquiteturais e de testes unitários que garantam a manutenibilidade e a estabilidade do código.

Tabela 2 - Requisitos Não Funcionais do SW

ID	Nome	Descrição
RNFSW01	Desempenho	A interface web deve refletir mudanças de temperatura em até 3 segundos após a publicação da métrica no <i>broker</i> MQTT.
RNFSW02	Segurança	Todas as rotas da API devem exigir autenticação JWT. As senhas devem ser armazenadas no banco de dados com criptografia bcrypt.
RNFSW03	Manutenibilidade	O código deve seguir o padrão REST, apresentar separação de responsabilidades (MVC Desacoplado) e estar devidamente documentado.
RNFSW04	Confiabilidade	O backend deve possuir cobertura de testes unitários focada nos Controladores (Controllers) através do framework Jest, garantindo a estabilidade e a validação das regras de negócio.

Fonte: Autora (2026).

6.1.3 Requisitos Funcionais do Sistema Embarcado

A Tabela 3 apresenta os Requisitos Funcionais do Sistema Embarcado, cujo objetivo é especificar as rotinas e lógicas de controle executadas pelo microcontrolador. Essas capacidades incluem o gerenciamento resiliente da conexão MQTT, a persistência de configurações, as regras de intertravamento dos equipamentos nos modos automático e manual, e o *feedback* sensorial de validação física das ações executadas na granja.

Tabela 3 - Requisitos Funcionais do SE

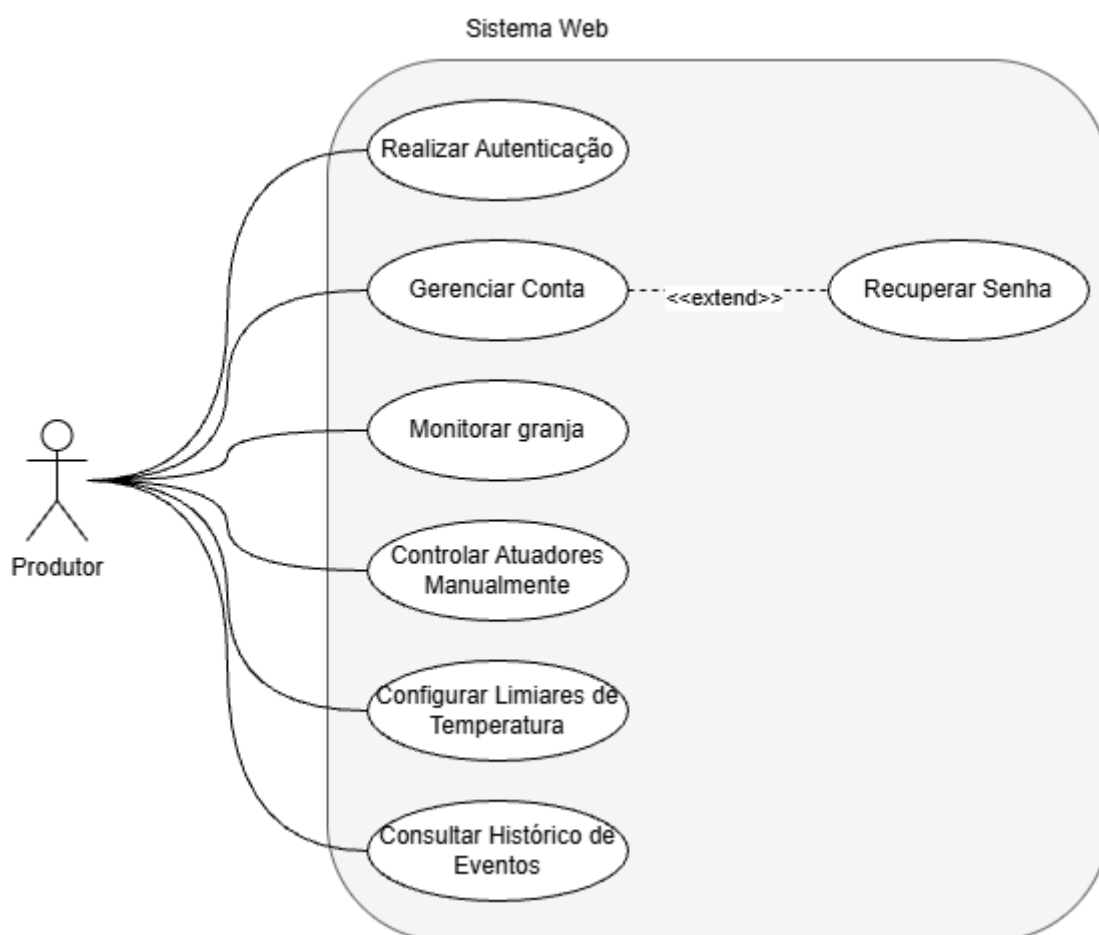
ID	Descrição	Prioridade
RFSE01	Conectar-se ao <i>broker</i> MQTT via Wi-Fi e autenticar-se com usuário e senha.	Alta
RFSE02	Ler a temperatura do sensor a cada 30 segundos e publicar no tópico <i>granja/climatizacao/temperatura</i> .	Alta

RFSE03	Receber e persistir em memória os limiares de temperatura enviados pelo sistema web.	Alta
RFSE04	Implementar reconexão automática ao <i>broker</i> MQTT em caso de perda de conexão, com backoff exponencial (1s, 2s, 4s... até 60s).	Alta
RFSE05	Publicar uma mensagem no tópico de <i>status</i> correspondente, com <i>payload</i> em formato JSON, contendo o estado atual, a origem (AUTO/MANUAL) e o <i>timestamp</i> , sempre que houver alteração no estado de qualquer atuador.	Alta
RFSE06	Operar em modo automático, obedecendo as regras: comparar a temperatura lida com os limiares configurados. Se $temp < temp_minima$ → ligar lâmpada e desligar ventilador. Se $temp > temp_maxima$ → ligar ventilador e desligar a lâmpada. Se $temp_minima \leq temp \leq temp_maxima$ → manter ambos atuadores desligados.	Alta
RFSE07	Executar a ação comandada e suspender o modo automático para o dispositivo em questão ao receber uma instrução via tópico MQTT direcionada a um atuador específico. A exclusão mútua deve ser garantida (ao ligar um componente, o outro deve ser desativado automaticamente).	Alta
RFSE08	Aguardar o intervalo de 3 segundos após o acionamento de qualquer relé para realizar a leitura do sensor LDR correspondente, publicando o resultado no tópico de <i>feedback</i> .	Alta

Fonte: Autora (2026).

6.2 Diagrama de Casos de Uso do Sistema Web

Figura 8 - Diagrama de Casos de Uso do SW



Fonte: Autora (2026).

6.2.1 Realizar Autenticação e Gerenciar Conta

Ator: Produtor.

Pré-condições: O produtor deve ter acesso à internet e possuir um e-mail válido previamente cadastrado.

Fluxo Principal:

1. O produtor acessa a tela de login.
2. Informa suas credenciais de acesso (e-mail/CPF/CNPJ e senha).
3. O sistema valida os dados e gera o *token* JWT.
4. O sistema redireciona o produtor ao painel de controle.

Fluxos Alternativos:

- A. **Cadastro de usuário:** O produtor preenche os dados cadastrais, o sistema criptografa a senha e salva no banco.
- B. **Recuperação de Senha:** O produtor solicita redefinição, o sistema envia um link por e-mail, e o usuário cadastra uma nova senha.

Pós-condições: Uma nova sessão segura é estabelecida ou uma nova conta é persistida no banco de dados.

6.2.2 Monitorar Granja

Ator: Produtor.

Pré-condições: O produtor deve estar autenticado. O hardware embarcado deve estar publicando dados no *broker* MQTT.

Fluxo Principal:

1. O produtor acessa o *Dashboard*.
2. O sistema recupera a última leitura de temperatura e o estado atual (ON/OFF) dos atuadores.
3. O sistema recupera as métricas de média de temperatura por período de tempo (7 dias/ 1 mês/ 3 meses).
4. O sistema recupera as últimas notificações enviadas ao produtor.

Pós-condições: O produtor visualiza as informações mais recentes da granja.

6.2.3 Controlar Atuadores Manualmente

Ator: Produtor.

Pré-condições: Produtor autenticado e dispositivo embarcado online e conectado ao *broker* MQTT.

Fluxo Principal:

1. O produtor seleciona o “Modo Manual”.
2. O produtor envia o comando para ligar ou desligar a lâmpada de aquecimento ou o ventilador.
3. O *backend* valida a regra de exclusão mútua para impedir o acionamento simultâneo de ambos dispositivos.
4. O sistema publica o comando no *broker* MQTT.
5. O sistema embarcado recebe a mensagem, aciona o módulo relé correspondente e realiza a leitura do sensor LDR para validação do *feedback* físico.

Pós-condições: O atuador físico muda de estado, a ação é registrada no histórico e uma notificação é enviada via Telegram.

6.2.4 Configurar Limiares de Temperatura

Ator: Produtor

Pré-condições: O produtor deve estar autenticado no sistema.

Fluxo Principal:

1. O produtor acessa a tela de configurações.
2. Insere os novos valores limites de temperatura mínima e máxima.
3. O sistema salva as informações no banco de dados MySQL.
4. O sistema publica as novas configurações via MQTT.
5. O sistema embarcado atualiza os parâmetros em sua memória local.

Pós-condições: O sistema embarcado passa a operar de forma automatizada com base nos novos limiares de temperatura configurados.

6.2.5 Consultar Histórico de Eventos

Ator: Produtor.

Pré-condições: O produtor deve estar autenticado no sistema.

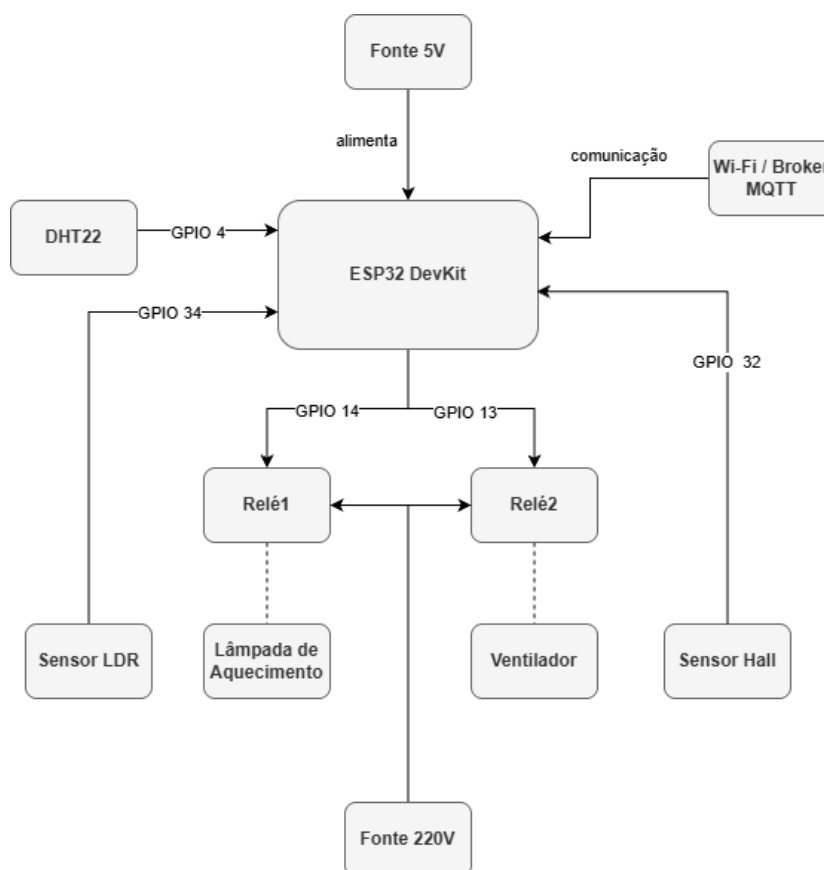
Fluxo Principal:

1. O produtor acessa a tela de histórico de eventos.
2. Aplica filtros de período de tempo, se desejar.
3. O sistema consulta o banco de dados.
4. A interface exibe a lista detalhando equipamento, ação, origem do comando, temperatura registrada e o *timestamp*.

Pós-condições: Os dados históricos são exibidos na interface de forma estruturada.

6.3 Diagrama de Blocos

Figura 9 - Diagrama de Blocos do SE



Fonte: Autora (2026).

O diagrama de blocos (Figura 9) ilustra a arquitetura de *hardware* do Sistema Embarcado, responsável pela coleta de dados do ambiente, controle dos atuadores físicos e comunicação direta com o SW via protocolo MQTT. O núcleo do processamento do sistema baseia-se no microcontrolador ESP32 DevKit, escolhido por sua capacidade de processamento e conectividade nativa Wi-Fi.

O ESP32 utiliza sua interface de rede sem fio para se conectar à internet e estabelecer comunicação bidirecional com o *broker* MQTT. Ele atua publicando dados de temperatura e status dos sensores/atuadores e escutando comandos de operação manual e novas configurações de limiares operacionais.

6.3.1 Entradas

Este bloco é responsável por monitorar as condições físicas da granja e o status dos equipamentos:

- Sensor DHT22: Atua como o sensor principal do sistema, sendo responsável por realizar as leituras de temperatura do ambiente. Esses dados são a base para a lógica de controle climático automático. O sensor DHT22 também é capaz de aferir a umidade do ar, porém essa métrica é omitida no presente projeto.
- Sensor LDR: Atua como um sensor de checagem luminosa. Ele é posicionado próximo à lâmpada de aquecimento para confirmar se o equipamento realmente acendeu quando comandado, permitindo ao sistema detectar falhas (como uma lâmpada queimada).
- Sensor Hall: Atua como um sensor de checagem magnética/mecânica. Sua função é monitorar a rotação física do ventilador, checando se o motor está em atividade após o seu acionamento pelo relé, auxiliando na detecção de falhas.

6.3.2 Saídas

Os atuadores são acionados por meio de módulos relé, que recebem o sinal lógico do ESP32 (3.3V) e fecham o circuito da rede de alimentação elétrica (220V) para energizar os equipamentos:

- Relé 1: Atua como interruptor de potência para a lâmpada de aquecimento. É acionado quando a temperatura declina abaixo do limiar mínimo configurado no sistema.
- Relé 2: Atua como interruptor de potência para o ventilador. É acionado quando a temperatura ultrapassa o limiar máximo.

6.3.3 Componentes de Hardware do Sistema Embarcado

Tabela 4 - Componentes de Hardware do SE

Componentes de Hardware	Modelo
Microcontrolador	ESP32 DevKit v1 (30 pinos)
Sensor de Temperatura	DHT22
Módulo Relé	Módulo relé 2 canais 5V
Sensor de luz	Sensor LDR Digital/ Analógico
Sensor de efeito hall	Sensor Hall
Protoboard + jumpers	-
Alimentação	5V

Fonte: Autora (2026).

7 RESULTADOS E DISCUSSÃO

7.1 Validação da API

Para assegurar a estabilidade do Sistema Web (SW), implementou-se uma suíte de 72 testes unitários utilizando o *framework* Jest. Conforme apresentado na Figura 3, o relatório de cobertura de código demonstra que os principais controladores do sistema atingiram cobertura superior a 90%, permitindo validar as regras de negócio relacionadas à autenticação, gerenciamento de limiares térmicos e controle dos atuadores.

A estratégia focou nos métodos da camada de controle, validando fluxos de sucesso e injeção de erros. As principais regras consolidadas foram:

- Autenticação e Cadastro: Verificação de credenciais criptografadas, geração de tokens, fluxos de recuperação de senha e consistência de dados (como validação de CPF/CNPJ e bloqueio de emails duplicados).
- Controle e Segurança de Atuadores: Confirmação de autorização de acesso aos dispositivos e validação do intertravamento lógico de segurança, garantindo que se um equipamento for ativado, o outro deve, obrigatoriamente, ser desativado, bloqueando requisições conflitantes.
- Monitoramento e Alertas: Processamento correto de leituras via MQTT e o disparo autônomo de notificações críticas de *hardware* (ex: falhas em lâmpadas de aquecimento ou falhas de rotação em motores).
- Parâmetros e Métricas: Bloqueio de configurações térmicas incoerentes (como temperatura mínima superior à máxima) e o cálculo preciso do tempo de uso dos equipamentos.

Como a abordagem focou na camada de controle utilizando simulações (*mocks*), as camadas de banco de dados (*models*) e comunicação externa (*services*) não foram executadas. Isso reflete a baixa cobertura nestes diretórios. Portanto, embora a lógica interna esteja validada, a comunicação física real com o banco MySQL, o *broker* MQTT e a API do Telegram escapa a este escopo.

7.2 Validação da Comunicação MQTT

Para atestar a confiabilidade da troca de mensagens entre o Sistema Embarcado e o Sistema Web, foram realizados testes de publicação (*publish*) e assinatura (*subscribe*) utilizando o protocolo MQTT. A validação teve como foco garantir o envio e o recebimento adequado de comandos de controle de estado (modo automático e manual), atualizações de limiares térmicos e o fluxo contínuo de leituras de temperatura e status dos sensores.

A comunicação foi estruturada em tópicos específicos. O SE publica ativamente suas métricas de *hardware* a cada 5 segundos no tópico *granja/suinos/status/disp_{id}*, o qual é escutado pelo *backend* (SW) através do serviço de mensageria. Em contrapartida, o SE assina o tópico *granja/suinos/config/disp_{id}* para receber ordens de controle e novas parametrizações. O uso da ferramenta Eclipse Mosquitto conectado ao MQTT Explorer, conforme já apresentado na Figura 4, permitiu inspecionar e validar o tráfego correto dessas mensagens em tempo real no *broker*.

Um ponto crítico validado nesta etapa foi a resiliência da entrega das atualizações dos limiares de temperatura através da rede. Como recurso de segurança e sincronização de estado, validou-se a publicação da nova configuração de temperatura pelo SW utilizando a *flag retain: true*.

O uso da retenção no *broker* MQTT garantiu o sucesso na sincronização assíncrona: caso o dispositivo embarcado perca a conexão com a *internet* ou esteja *offline* no exato momento da alteração via plataforma *web*, a mensagem fica armazenada no servidor. Dessa forma, a placa recebe a parametrização mais recente de forma automática e imediata assim que restabelece o *link* de rede e renova a assinatura no tópico.

A Tabela 5 mostra os cenários de comunicação testados e seus respectivos resultados:

Tabela 5 - Cenários de teste de validação da comunicação MQTT

Cenário	Resultado
Publicação de leitura de temperatura e status (SE)	Sucesso
Envio de comando de controle de atuadores (SW)	Sucesso
Recebimento e execução do comando de controle (SE)	Sucesso

Publicação de novos limiares de temperatura com retenção (SW)	Sucesso
Recebimento da atualização de temperatura (SE)	Sucesso

Fonte: Autora (2026).

7.3 Validação do Sistema Embarcado

A validação do Sistema Embarcado (SE) em ambiente de bancada atestou a estabilidade do microcontrolador ESP32 e o correto acionamento dos componentes físicos operando em lógica *Active-Low*. O sensor DHT22 demonstrou precisão na captação térmica respeitando o intervalo de leitura de 5 segundos. Para o controle de ambiência, utilizou-se um relé de dois canais: o acionamento da ventilação foi atestado utilizando uma ventoinha real, sem o acoplamento de uma lâmpada física como carga térmica.

Os mecanismos de segurança e integridade de *hardware* também foram validados mediante simulações controladas. A capacidade do sensor LDR em alertar falhas na iluminação de aquecimento foi testada com sucesso utilizando a lanterna de um aparelho celular. Simultaneamente, o sensor magnético Hall registrou os pulsos mecânicos da ventoinha em funcionamento, garantindo a rápida identificação de eventuais paradas ou travamentos do motor.

Por fim, ensaios operacionais comprovaram a reconexão autônoma da placa ao MQTT após quedas de energia, bem como a recuperação imediata e bem sucedida dos limiares térmicos, que permaneceram salvos na memória Flash não volátil do microcontrolador (via biblioteca Preferences).

A Tabela 6 consolida os testes realizados no protótipo.

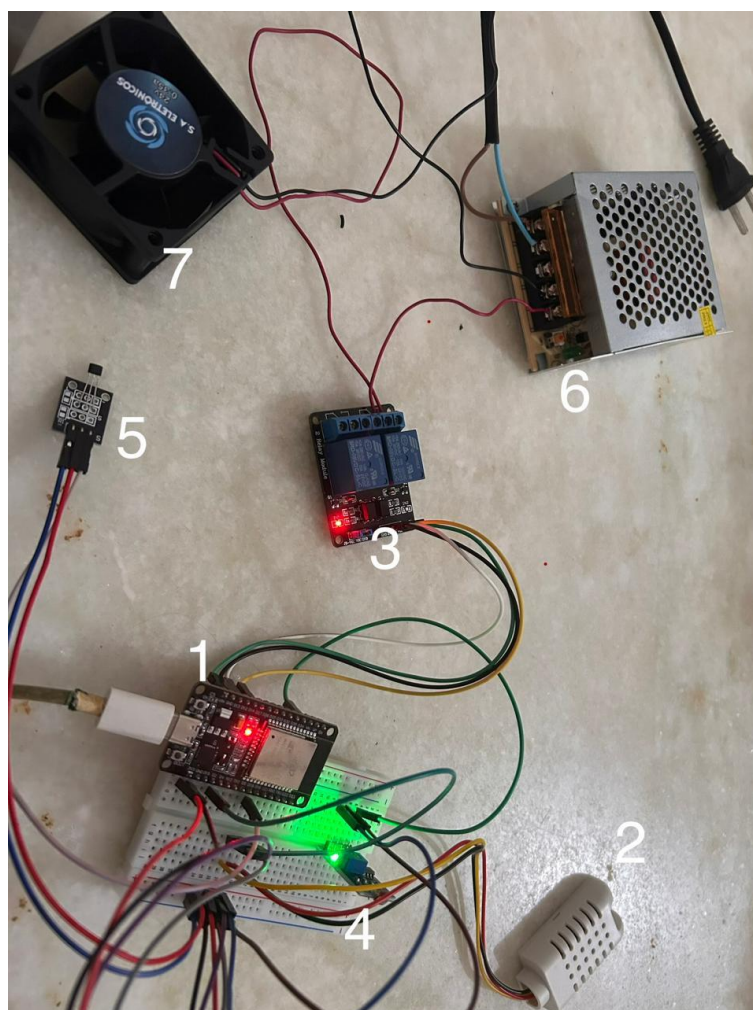
Tabela 6 - Cenários de Teste no Protótipo

Cenário	Resultado
Leitura ambiental via sensor DHT22	Aprovado
Comutação do relé para aquecimento (sem carga física)	Aprovado
Acionamento e intertravamento lógico da ventoinha	Aprovado
Deteção de falha no aquecimento (LDR simulado com lanterna)	Aprovado
Deteção de rotação mecânica (Sensor Hall na ventoinha)	Aprovado
Persistência e recuperação de limiares via memória Flash	Aprovado
Reconexão autônoma dos serviços de rede (MQTT)	Aprovado

Fonte: Autora (2026).

A Figura 10 mostra o protótipo do *hardware*.

Figura 10 - Protótipo de *hardware* estágio final



Fonte: Autora (2026).

1: ESP32. 2: DHT22. 3: Módulo relé de dois canais. 4: Sensor LDR. 5: Sensor Hall. 6: Fonte Chaveada. 7: Ventoinha.

7.4 Testes Integrados

Enquanto as validações anteriores garantiram o funcionamento isolado de cada camada, os testes integrados (*End-to-End*) avaliaram o fluxo completo da informação na arquitetura. O objetivo foi validar a reação em cadeia da solução: desde a captação física no ambiente, passando pelo tráfego via MQTT e processamento no *backend*, até a notificação final na *interface* web e no *smartphone* do usuário.

Os cenários a seguir descrevem as respostas da arquitetura unificada operando em conjunto:

Cenário 1: Queda de temperatura (Abaixo do limiar mínimo)

- Ação inicial: O sensor DTH22 realiza a leitura e a ESP32 identifica que a temperatura ambiente cai para um valor inferior à *temp_minima* configurada.
- Resultado esperado: O microcontrolador (SE) deve acionar o relé da lâmpada de aquecimento e desligar o ventilador (caso estivesse ligado). O SE deve publicar a mudança de estado no MQTT. O *backend* (SW) deve interceptar a mensagem, persistir o evento no banco de dados e disparar uma notificação via Telegram.
- Resultado observado: Comportamento correto. A cadeia de automação e notificação ocorreu em tempo real sem falhas de sincronia.

Cenário 2: Elevação de temperatura (Acima do limiar máximo)

- Ação inicial: A temperatura do ambiente ultrapassa o limiar da *temp_maxima*.
- Resultado esperado: Seguindo a lógica de intertravamento e histerese, o SE deve acionar o ventilador e garantir que a lâmpada seja desativada. O *status* atualizado é publicado via MQTT. O SW processa a exclusão mútua, armazena o histórico e notifica o produtor sobre o acionamento do modo de resfriamento.
- Resultado observado: Comportamento correto.

Cenário 3: Sobrescrita de controle manual remoto

- Ação inicial: O produtor, autenticado no Sistema Web, aciona o botão para ligar o ventilador remotamente.
- Resultado esperado: O SW publica um *payload* MQTT com a diretriz *modo: MANUAL* e o *ventilador: ON*. O SE recebe o comando, suspende a rotina automática, aciona o relé do ventilador e publica o *feedback* de confirmação. O SW registra o evento de origem manual no banco e envia um alerta ao Telegram informando a intervenção. Após o esgotamento do *timeout* de segurança, configurado em 1 minuto estritamente para viabilizar os testes de

bancada, mas projetado para 2 horas no ambiente de produção real, o SE deve retornar automaticamente ao modo *AUTO*.

- Resultado observado: Comportamento correto, incluindo o retorno temporizado autônomo ao modo automático pelo sistema embarcado.

Cenário 4: Integração de Alerta Crítico (Falha de *Hardware*)

- Ação inicial: O sistema entra em estado de aquecimento (relé da lâmpada em ON), mas o sensor LDR acoplado indica ausência de luminosidade (*luz_detectada: NAO*).
- Resultado esperado: O SE detecta a incoerência física e anexa o alerta de falha no *payload* MQTT. O SW lê a *flag* de aviso, identifica a gravidade do evento, insere o registro crítico no histórico de notificações do banco de dados e aciona o *TelegramService* enviando a mensagem formatada de emergência.
- Resultado observado: Comportamento correto, validando a segurança de ponta a ponta da solução.

7.5 Discussão dos Resultados

Os resultados obtidos demonstram que a arquitetura proposta foi capaz de atender aos requisitos definidos para o Sistema de Automação do Manejo Térmico (SAMT). A integração entre o Sistema Embarcado (SE) e o Sistema Web (SW) ocorreu de forma satisfatória, permitindo o fluxo contínuo de informações entre os sensores instalados no ambiente, os mecanismos de controle dos atuadores e a interface utilizada pelo produtor.

A utilização do protocolo MQTT mostrou-se compatível com os requisitos do projeto. Sua arquitetura baseada em publicação e assinatura permite desacoplar as diferentes camadas do sistema, reduzindo a dependência direta entre os componentes e facilitando a troca de mensagens em tempo real. Durante os testes realizados, não foram observadas inconsistências na sincronização dos estados dos dispositivos ou na atualização dos dados apresentados ao usuário.

O microcontrolador ESP32 apresentou desempenho compatível com as necessidades da solução. Além de disponibilizar conectividade Wi-Fi integrada, o dispositivo foi capaz de executar simultaneamente as rotinas de leitura dos sensores, processamento das regras de automação, gerenciamento da comunicação MQTT e acionamento dos atuadores. Esse resultado reforça a viabilidade da utilização de plataformas de baixo custo em aplicações de automação voltadas ao agronegócio.

Outro aspecto relevante identificado durante a validação foi a implementação dos mecanismos de *feedback* físico por meio dos sensores LDR e Hall. Diferentemente de sistemas que apenas assumem que o comando enviado ao atuador foi executado corretamente, a solução desenvolvida incorpora uma camada adicional de verificação, capaz de detectar situações de falha física nos equipamentos monitorados. Essa característica contribui para aumentar a confiabilidade operacional do sistema e possibilita a geração de alertas em tempo real para o produtor.

Comparando-se o SAMT com os trabalhos relacionados apresentados neste estudo, observa-se que a solução proposta amplia significativamente as capacidades de monitoramento e gerenciamento remoto. Enquanto os trabalhos analisados concentram-se principalmente no controle local dos equipamentos ou no armazenamento dos dados em mídias físicas, o sistema desenvolvido disponibiliza uma interface web centralizada, persistência de dados em banco relacional, notificações instantâneas e parametrização remota dos limiares de temperatura. Essas funcionalidades proporcionam maior praticidade operacional e ampliam a capacidade de acompanhamento do ambiente produtivo.

Sob a perspectiva da Engenharia de Software, a adoção de uma arquitetura desacoplada baseada em API REST contribuiu para a organização e manutenção do sistema. A separação entre *frontend*, *backend* e sistema embarcado permite que futuras evoluções sejam realizadas de forma independente, reduzindo o impacto de modificações em componentes específicos da solução. Essa característica favorece a escalabilidade do projeto e possibilita a incorporação de novos recursos sem necessidade de reestruturações significativas.

Apesar dos resultados positivos, algumas limitações devem ser consideradas. Os testes foram conduzidos em ambiente de bancada e em condições controladas, não contemplando fatores normalmente encontrados em instalações produtivas reais, com interferências eletromagnéticas, falhas de conectividade e variações climáticas extremas. Dessa forma, embora o sistema tenha demonstrado estabilidade durante as validações realizadas, sua robustez em cenários de operação contínua ainda necessita de avaliação complementar.

Por fim, os resultados obtidos indicam que a arquitetura proposta é tecnicamente viável e possui potencial para aplicação em propriedades de pequeno e médio porte. A combinação entre sistemas embarcados, computação web e tecnologias de Internet das Coisas demonstrou ser capaz de fornecer uma solução funcional para monitoramento e automação do manejo térmico, constituindo uma base sólida para futuras expansões e validações em ambientes produtivos reais.

7.6 Limitações do Estudo

Os resultados apresentados nas seções anteriores foram obtidos integralmente em ambiente de bancada, sob condições controladas de laboratório, e não em uma instalação suinícola real. Essa distinção é relevante para a correta interpretação do alcance dos testes realizados. O protótipo não foi submetido a variações climáticas extremas (como altas temperaturas e umidade elevada, características do ambiente real de uma granja), tampouco a cenários de instabilidade prolongada de conectividade Wi-Fi ou de interferências eletromagnéticas que podem ocorrer em um galpão de produção, onde a presença de motores, equipamentos elétricos de maior porte e estruturas metálicas tende a impactar de forma mais significativa a comunicação sem fio e o funcionamento dos componentes eletrônicos do que em bancada.

Outro ponto que constitui uma limitação direta deste estudo é o posicionamento físico dos sensores de verificação. O arranjo do sensor LDR em relação à lâmpada de aquecimento e do sensor Hall em relação à ventoinha foi definido especificamente para as condições do protótipo de bancada, com os componentes montados em curta distância e em ambiente sem outras fontes de luz ou de campo magnético que pudessem interferir nas leituras. Em uma instalação

real, o posicionamento desses sensores em relação aos atuadores é um fator crítico para o correto funcionamento do sistema de verificação: a incidência de luz natural ou artificial proveniente de outras fontes poderia comprometer a leitura do sensor LDR, assim como a distância e o alinhamento do sensor Hall em relação às pás ou ao eixo da ventoinha real (de maior porte que a utilizada nos testes) exigiriam recalibração e possivelmente um novo estudo de posicionamento. Dessa forma, a transposição do protótipo para um ambiente produtivo real demandaria uma etapa adicional de engenharia para redefinir a disposição física dos componentes de hardware, o que é indicado como uma limitação do presente trabalho e como uma recomendação para pesquisas futuras.

Por fim, cabe destacar que os cenários de teste da comunicação MQTT e da validação do protótipo, apresentados nas Tabelas 5 e 6, foram reportados de forma binária (sucesso/aprovado), sem a coleta de métricas quantitativas de desempenho, como latência de entrega das mensagens, taxa de perda de pacotes ou desvio de precisão do sensor DHT22 em relação a um instrumento de referência calibrado. Os testes também foram conduzidos com um único dispositivo embarcado publicando no broker, não tendo sido avaliado o comportamento do sistema diante da publicação simultânea de múltiplos dispositivos. A ausência dessas medições quantitativas e dos testes de carga limita a generalização dos resultados para cenários de operação contínua e em maior escala, sendo recomendada, como trabalho futuro, a condução de testes de campo instrumentados com métricas de desempenho e de carga que permitam validar a robustez do sistema em condições reais de produção.

8 CONSIDERAÇÕES

O projeto Sistema de Automação do Manejo Térmico buscou ser uma ferramenta para auxiliar no desenvolvimento e eficiência da cadeia produtiva da suinocultura, tendo em vista as possibilidades de integração tecnológica no setor e a importância do cuidado com a ambiência na qual o rebanho está inserido.

A concepção dessa solução permitiu a aplicação prática de diversas técnicas e tecnologias aprendidas ao longo do curso de Análise e Desenvolvimento de Sistemas, resultando em um produto capaz de mitigar problemas reais, inclusive no âmbito regional.

Conforme discutido na seção 7.6, os testes realizados neste trabalho concentraram-se em ambiente de bancada, sob condições controladas, o que é adequado à etapa de prova de conceito do projeto, mas não substitui a validação em campo. Ainda assim, observa-se que a Internet das Coisas (IoT) é multifacetada e possui capacidade para integrar diferentes cenários, desde *smartwatches* e casas inteligentes até a pecuária. De forma complementar, os sistemas web são amplamente consolidados em diversos setores industriais. A convergência de ambas as tecnologias não apenas permitiu a consolidação dos conhecimentos acadêmicos da autora na criação de um sistema robusto, mas também evidenciou o potencial técnico para agregar valor real ao agronegócio.

Como proposta para trabalhos futuros, além da aplicação de modelos de Inteligência Artificial (IA) para a geração automatizada de relatórios gerenciais e de modelos de Aprendizado de Máquina (*Machine Learning*) para controle preditivo do microclima a partir de dados históricos da granja, destacam-se outras frentes de evolução identificadas a partir das limitações apontadas na seção 7.6: (i) a validação do protótipo em ambiente produtivo real, com o correspondente reposicionamento e recalibração dos sensores de verificação física (LDR e Hall) em relação aos atuadores; (ii) a condução de testes de carga com múltiplos dispositivos embarcados publicando simultaneamente no broker MQTT, cenário já previsto na modelagem de dados do sistema; e (iii) a coleta de métricas quantitativas de desempenho, como latência de entrega de mensagens, taxa de perda de pacotes e precisão do sensor DHT22 frente a instrumentos de referência calibrados.

REFERÊNCIAS

AOSONG ELECTRONICS. **DHT22 (AM2302) Digital-output relative humidity & temperature sensor/module**: datasheet. [S. l.], [s. d.]. Disponível em: <https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf>. Acesso em: 18 ago. 2026.

ASSOCIAÇÃO BRASILEIRA DOS CRIADORES DE SUÍNOS (ABCS). **Retrato da Suinocultura Brasileira**. 2024. Disponível em: <https://abcs.org.br/wp-content/uploads/2024/04/Retrato-da-Suinocultura-2024-Web.pdf>. Acesso em: 04 jun. 2026.

BEKER, F. M. **Ambiência em Suínos**: do controle da temperatura a suinocultura 4.0. In: PEREIRA, A. M. *et al.* Investigação Científica e Técnica em Medicina Veterinária 2. Belo Horizonte: Atena Editora, 2020.

BORGES, Pedro Hurtado de Mendoza; MENDOZA, Zaíra Morais dos Santos Hurtado; MORAIS, Pedro Hurtado de Mendoza; DOS SANTOS, Ronei Lopes. Sistema Automatizado de Baixo Custo para Produtores Rurais: controle e monitoramento do ambiente térmico na suinocultura. **Revista Eletrônica Competências Digitais para Agricultura Familiar**, Tupã, São Paulo, Brasil, v. 4, n. 2, p. 177–199, 2018. Disponível em: <https://owl.tupa.unesp.br/recodaf/index.php/recodaf/article/view/79>. Acesso em: 22 jun. 2026.

ESPRESSIF SYSTEMS. **ESP32 Series Datasheet**. [S. l.], 2024. Disponível em: https://documentation.espressif.com/esp32_datasheet_en.pdf. Acesso em: 16 ago. 2026.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. Tese (Doutorado em Ciência da Computação) - University of California, Irvine, Irvine, 2000. Disponível em: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>. Acesso em: 18 ago. 2026.

FILHO, J. R. S. V. **Setor de Suinocultura em Pernambuco**. ADEPE, 2025. Disponível em: <https://www.adepe.pe.gov.br/wp-content/uploads/2025/02/Analise-do-setor-de-Suinocultura.pdf>. Acesso em: 03 jun. 2026.

Folha de Pernambuco. **Suinocultores do Agreste apostam em práticas sustentáveis para aumentar produtividade**. 2025. Disponível em: <https://www.folhape.com.br/economia/suinocultores-do-agreste-apostam-em-praticas-sustentaveis-para/419764/>. Acesso em: 03 jun. 2026.

GRPC. What is gRPC? [S. l.], [s. d.]. Disponível em: <https://grpc.io/docs/what-is-grpc/introduction/>. Acesso em: 18 ago. 2026.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **Rebanho de Suínos (Porcos)**. 2024. Disponível em: <https://www.ibge.gov.br/explica/producao-agropecuaria/suinos/pe>. Acesso em: 02 jun. 2026.

IBGE. **Produção da Pecuária Municipal 2024 (PPM)**. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/periodicos/84/ppm_2024_v52_br_informativo.pdf. Acesso em: 04 jun. 2026.

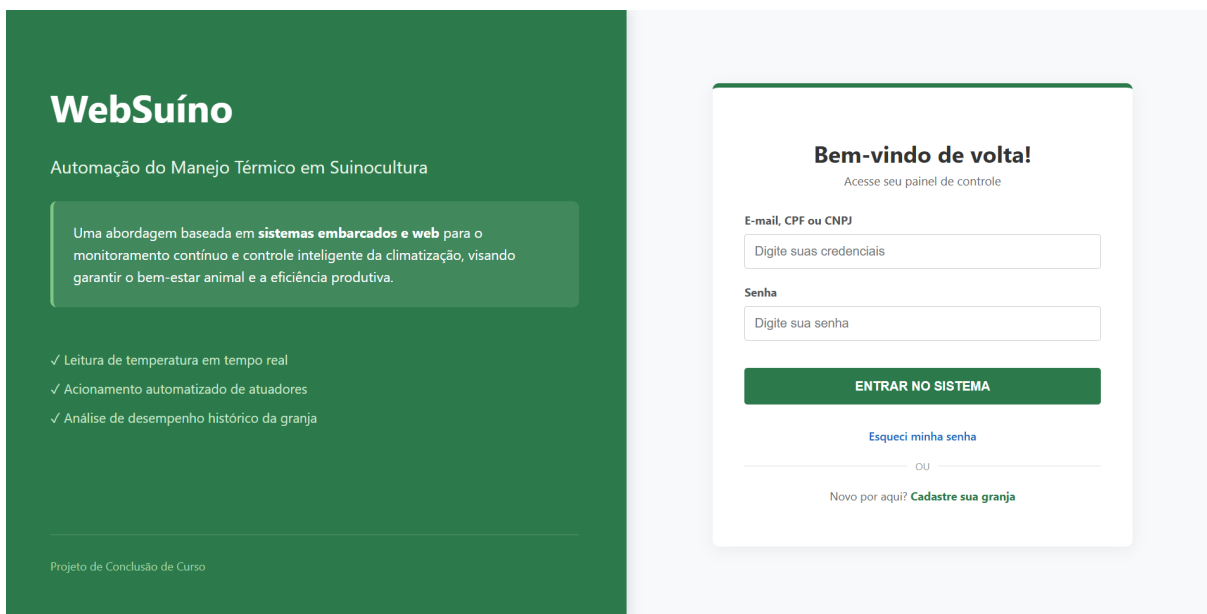
MIELE, M.; TALAMINI, D. J. D.; MARTINS, F. M.; SANDI, A. J. **Indicadores da suinocultura brasileira**. In: BEM-ESTAR animal e sustentabilidade na suinocultura: implementando a IN 113. Brasília, DF: ABCS, 2025. p. 15-23.

MQTT.ORG. **MQTT**: The Standard for IoT Messaging. [S. l.], [s. d.]. Disponível em: <https://mqtt.org/>. Acesso em: 16 ago. 2026.

SILVA, Eduardo Torres Mendes da; SOUZA, Silvia Regina Lucas de; COSTA, Nathiely Ziviani; OLIVEIRA, Kassandra Sussi Mustafé; DUARTE, William Bailo. SISTEMA DE MONITORAMENTO DE TEMPERATURA E UMIDADE DE BAIXO CUSTO PARA MATERNIDADE DE SUÍNOS. **ENERGIA NA AGRICULTURA**, [S. l.], v. 37, n. 2, p. 1–11, 2022. DOI: 10.17224/EnergAgric.2022v37n2p1-11. Disponível em: <https://revistas.fca.unesp.br/index.php/energia/article/view/4483>. Acesso em: 22 jun. 2026.

APÊNDICE A - TELAS DA INTERFACE DO SISTEMA WEB

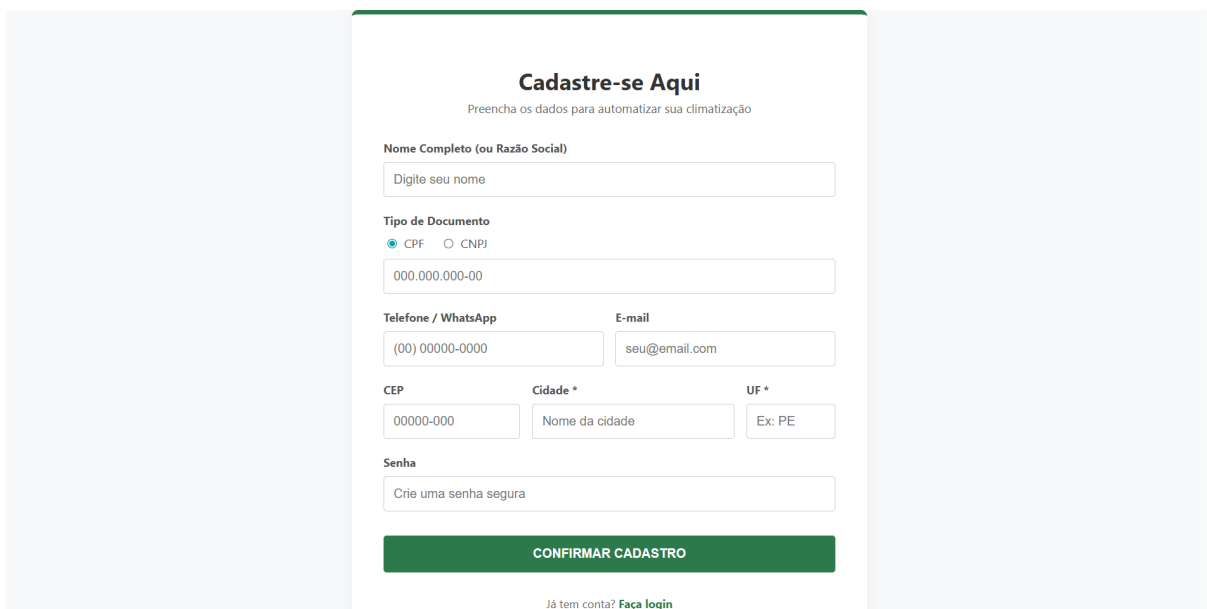
Figura 11 - Tela Inicial



A tela inicial do sistema WebSuíno é dividida em duas seções principais. À esquerda, um painel verde escuro com o título "WebSuíno" em branco. Abaixo do título, o subtítulo "Automação do Manejo Térmico em Suinocultura" é exibido. Um box verde claro contém o texto: "Uma abordagem baseada em **sistemas embarcados e web** para o monitoramento contínuo e controle inteligente da climatização, visando garantir o bem-estar animal e a eficiência produtiva." Abaixo, há uma lista de funcionalidades com marcadores de verificação: "✓ Leitura de temperatura em tempo real", "✓ Acionamento automatizado de atuadores" e "✓ Análise de desempenho histórico da granja". No rodapé, o texto "Projeto de Conclusão de Curso" é visível. À direita, uma caixa branca com sombra contém o título "Bem-vindo de volta!" e o subtítulo "Acesse seu painel de controle". Abaixo, há campos para "E-mail, CPF ou CNPJ" e "Senha", ambos com o placeholder "Digite suas credenciais". Um botão verde "ENTRAR NO SISTEMA" está abaixo dos campos. Links para "Esqueci minha senha" e "Novo por aqui? Cadastre sua granja" são exibidos no rodapé da caixa.

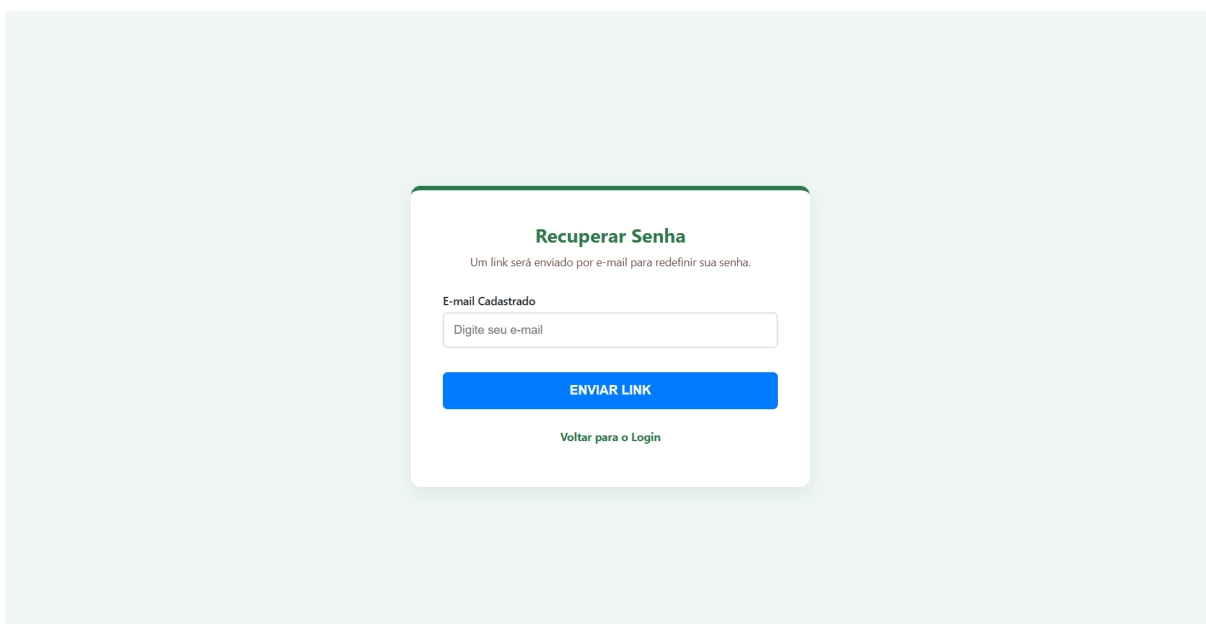
Fonte: Autora (2026).

Figura 12 - Tela de Cadastro



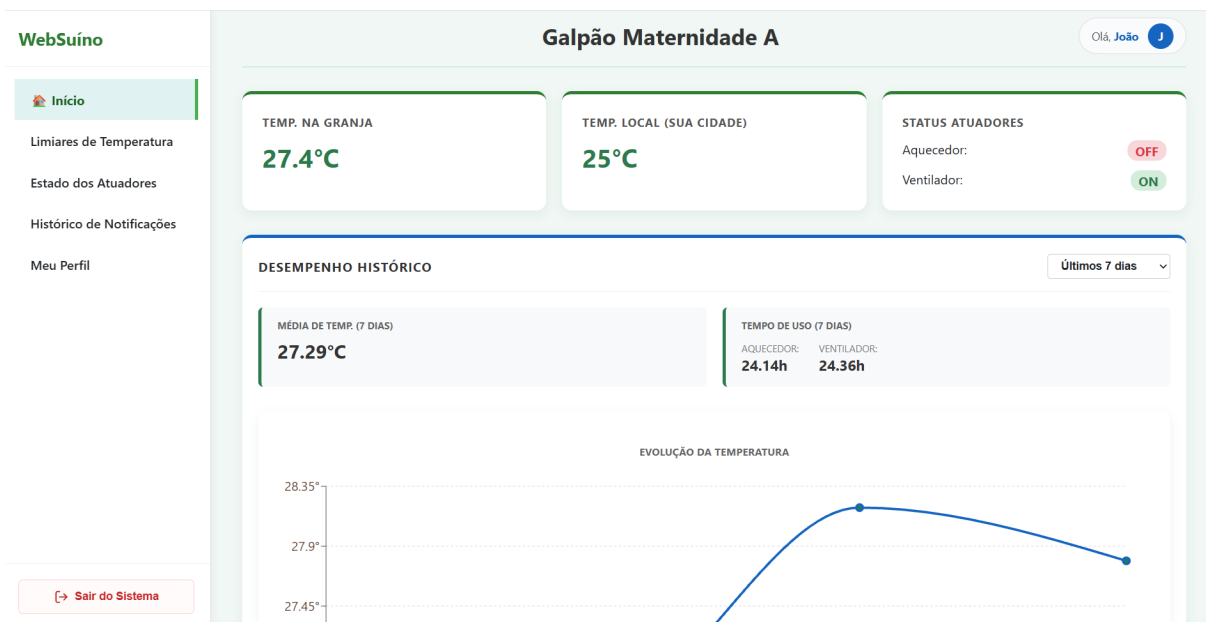
A tela de cadastro do sistema WebSuíno é uma caixa branca com sombra. No topo, o título "Cadastre-se Aqui" é exibido, seguido pelo subtítulo "Preencha os dados para automatizar sua climatização". Os campos de formulário são: "Nome Completo (ou Razão Social)" com o placeholder "Digite seu nome"; "Tipo de Documento" com radio buttons para "CPF" (selecionado) e "CNPJ", e um campo para o número do documento com o placeholder "000.000.000-00"; "Telefone / WhatsApp" com o placeholder "(00) 00000-0000" e "E-mail" com o placeholder "seu@email.com"; "CEP" com o placeholder "00000-000", "Cidade *" com o placeholder "Nome da cidade" e "UF *" com o placeholder "Ex: PE"; e "Senha" com o placeholder "Crie uma senha segura". Um botão verde "CONFIRMAR CADASTRO" está no rodapé da caixa. Abaixo do botão, o texto "Já tem conta? Faça login" é exibido.

Fonte: Autora (2026).

Figura 13 - Tela de Recuperação de Senha

A interface de recuperação de senha é apresentada em uma caixa centralizada sobre um fundo verde claro. No topo, o título "Recuperar Senha" é exibido em verde, seguido por uma instrução: "Um link será enviado por e-mail para redefinir sua senha." Abaixo, há um campo de entrada rotulado "E-mail Cadastrado" com o placeholder "Digite seu e-mail". Um botão azul com o texto "ENVIAR LINK" está posicionado logo abaixo do campo. Na base da caixa, há um link "Voltar para o Login" em verde.

Fonte: Autora (2026).

Figura 14 - Tela de Início do Sistema

Fonte: Autora (2026).

Figura 15 - Tela de Início do Sistema (Continuação)



Fonte: Autora (2026).

Figura 16 - Tela de Configuração de Limiares

A interface de configuração de clima da WebSuíno permite definir os limites de temperatura para o sistema. O formulário 'LIMIARES DA GRANJA' inclui campos para a temperatura mínima (22.00°C) e máxima (26.00°C), com explicações sobre quando o aquecedor e o ventilador serão ativados. Um botão 'Salvar Configurações' está no final.

Configuração Atual (Mínima):	Configuração Atual (Máxima):
22.00°C	26.00°C

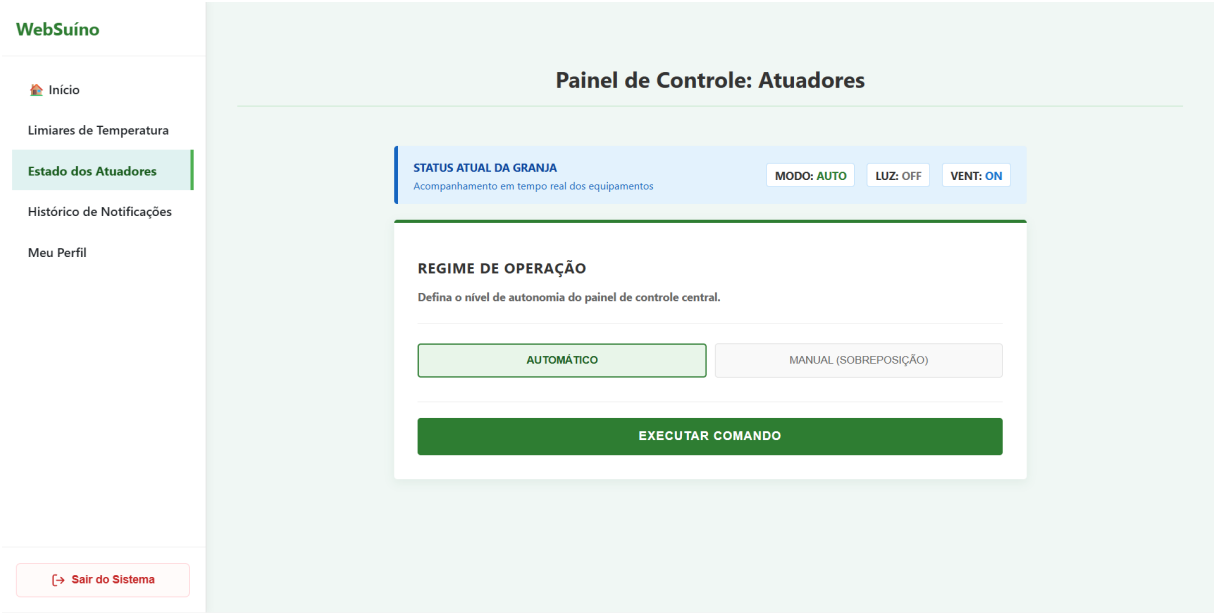
Temperatura Mínima (°C): 22.00
Abaixo deste valor, o Aquecedor será ligado.

Temperatura Máxima (°C): 26.00
Acima deste valor, o Ventilador será ligado.

Salvar Configurações

Fonte: Autora (2026).

Figura 17 - Tela de Controle dos Atuadores - Modo Automático



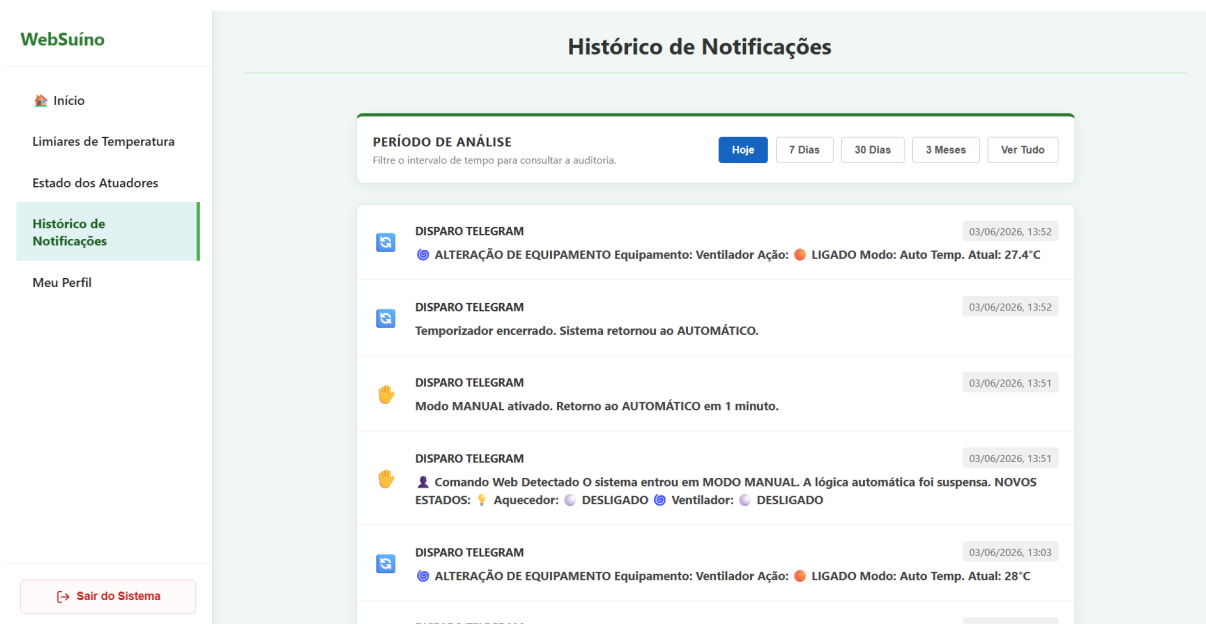
Fonte: Autora (2026)

Figura 18 - Tela de Controle dos Atuadores - Modo Manual



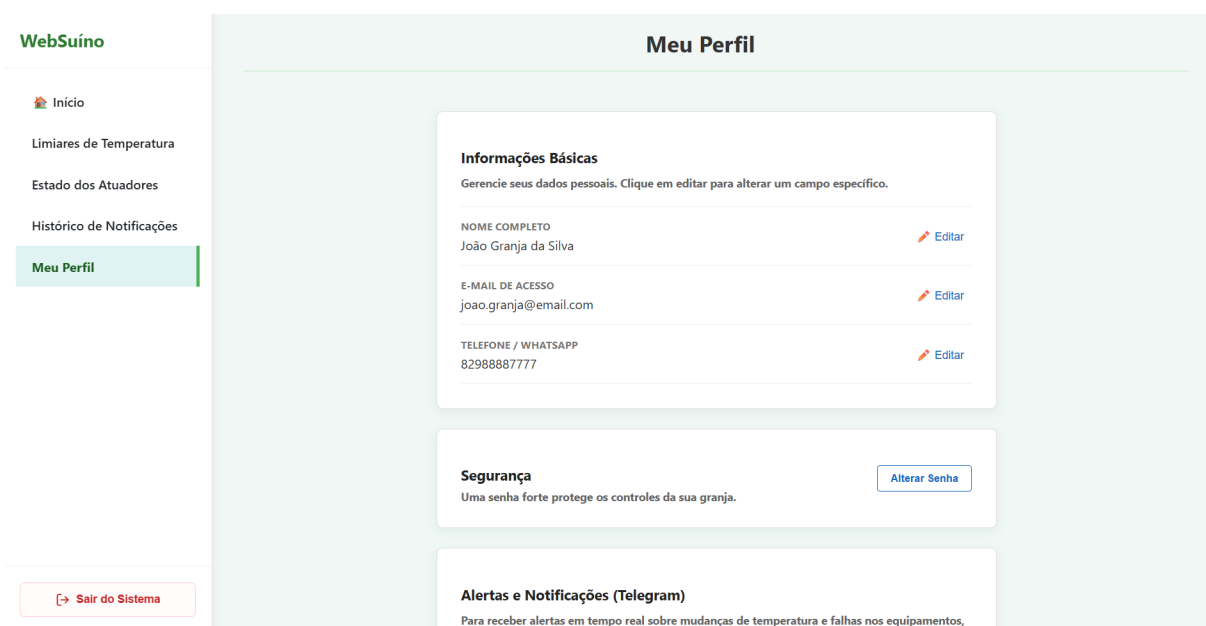
Fonte: Autora (2026).

Figura 19 - Tela de Histórico de Notificações



Fonte: Autora (2026).

Figura 20 - Tela de Informações do Perfil



Fonte: Autora (2026).

Figura 21 - Tela de Informações do Perfil (Continuação)

WebSuíno

🏠 Início

🌡️ Limiares de Temperatura

🔌 Estado dos Atuadores

📜 Histórico de Notificações

👤 Meu Perfil

➔ Sair do Sistema

Segurança
Uma senha forte protege os controles da sua granja.

Senha Atual

Nova Senha

Confirmar Nova Senha

Cancelar

Salvar Senha

Alertas e Notificações (Telegram)
Para receber alertas em tempo real sobre mudanças de temperatura e falhas nos equipamentos, cadastre o seu ID do Telegram.

Como descobrir seu ID: Abra o Telegram, envie uma mensagem para o bot @userinfobot e cole o número recebido abaixo.

ID DO CHAT (TELEGRAM)

5523210414

✎ Editar

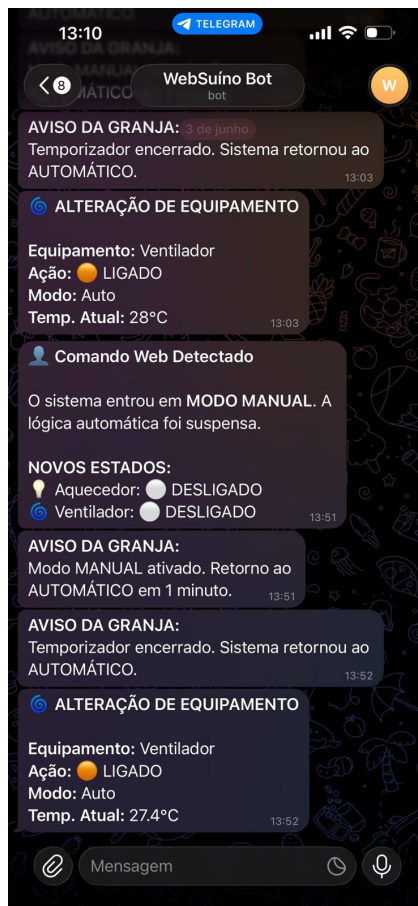
Excluir Conta
Remova permanentemente seus dados do sistema.

DELETAR

Fonte: Autora (2026).

APÊNDICE B - TELAS DE NOTIFICAÇÕES

Figura 22 - Tela do Chat do Telegram com Notificações



Fonte: Autora (2026).

Figura 23 - Notificação de Recuperação de Senha



Fonte: Autora (2026).

APÊNDICE C - DIAGRAMA ENTIDADE RELACIONAMENTO EM NOTAÇÃO CROW’S FOOT

Figura 24 - DER em Crow’s Foot

