



Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande

Coordenação do Curso Superior Bacharelado em Engenharia de
Computação

Colmeia digital: Sistema de Gestão e Monitoramento de Colmeias de Abelhas Apis com ESP32

Arlan da Silva Santos

Diogo da Silva Santos

Orientador: Prof. Dr. Moacy Pereira da Silva



Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Curso Superior Bacharelado em Engenharia de
Computação

Colmeia digital: Sistema de Gestão e Monitoramento de Colmeias de Abelhas Apis com ESP32

Arlan da Silva Santos
Diogo da Silva Santos

Trabalho de conclusão de curso apresentado
ao Curso de Bacharelado em Engenharia de
Computação, do Instituto Federal de Educação,
Ciência e Tecnologia da Paraíba - *Campus*
Campina Grande, em cumprimento às exigências
parciais para a obtenção do título de Bacharel
em Engenharia de Computação.

Orientador: Prof. Dr. Moacy Pereira da Silva

Campina Grande, dezembro de 2025

Catálogo na fonte:

Ficha catalográfica elaborada por Gustavo César Nogueira da Costa - CRB 15/479

S237c Santos, Arlan da Silva

Colmeia digital: sistema de gestão e monitoramento de colmeias de abelhas Apis com ESP32 / Arlan da Silva Santos; Diogo da Silva Santos. - Campina Grande, 2025.
82f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Bacharelado em Engenharia de Computação) - Instituto Federal da Paraíba, 2025.

Orientador: Prof. Dr. Moacy Pereira da Silva

1. Engenharia de Computação 2. Apicultura 3. Sistemas embarcados. 4. Internet das coisas. 5. Monitoramento remoto I. Santos, Diogo da Silva II. Silva, Moacy Pereira da III. Título.

CDU 004.4:638

Arlan da Silva Santos
Diogo da Silva Santos

Prof. Dr. Moacy Pereira da Silva

Prof. Dr. Elmano Ramalho Cavalcanti
Membro da Banca

Prof. Dr. José Antonio Candido Borges da Silva
Membro da Banca

Campina Grande, Paraíba, Brasil
Dezembro de 2025

"Os que confiam no senhor são como os montes de Sião."
(Salmo 125:1)

Agradecimentos

Por Arlan: Primeiramente, agradeço a Deus, a quem dedico toda honra e glória por me dar força e sabedoria ao longo desta jornada.

Aos meus avós, que me criaram e me ofereceram toda a estrutura necessária para que eu chegasse até aqui. Em especial, dedico um agradecimento profundo à minha avó Margarida, por todo o seu incentivo incansável aos meus estudos e por sempre acreditar em mim. A toda a minha família, meu muito obrigado pelo amor, apoio e compreensão nos momentos de ausência.

A todos os professores que, ao longo do curso, compartilharam seus valiosos conhecimentos e contribuíram para a minha formação. Em particular, agradeço ao nosso orientador, Professor Moacy, pela paciência, direcionamento e por acreditar no potencial deste projeto desde o início.

Aos meus amigos e colegas, pela parceria, pelo companheirismo e pelos momentos de apoio que foram fundamentais durante todo o percurso, em especial ao meu parceiro de trabalho, Diogo. Sua dedicação, amizade e colaboração foram essenciais para superarmos os desafios e concluirmos este projeto com sucesso. Muito obrigado pela jornada.

A todos que, de alguma forma, tornaram a realização deste trabalho possível, a minha sincera gratidão.

Por Diogo: Agradeço ao criador, pelo sopro de vida, proteção e discernimento.

A meus pais e avós, pelo cuidado e direcionamento de que os estudos são a melhor forma de alcançar uma vida boa, próspera e justa. Que nunca deixaram faltar a estrutura para que fosse possível concluir o curso técnico e graduação.

A instituição em que fui acolhido desde o ano de 2016 ao ingressar no técnico integrado ao ensino médio, onde firmei meus alicerces no conhecimento e onde criei os melhores laços.

Ao professor e orientador Dr. Moacy Pereira, que tive contato desde o primeiro semestre, onde me transmitiu muito conhecimento.

A meu colega de curso e carreira, Arlan, por sua amizade, cooperação em toda jornada de curso e por ter acreditado no potencial do projeto e muito contribuir para o crescimento do mesmo.

Resumo

É apresentado no presente trabalho o desenvolvimento de um sistema embarcado inteligente de ponta a ponta, projetado para a gestão e o monitoramento remoto de colmeias de abelhas da espécie *Apis mellifera* em apicultura racional. Integra-se à solução um nó sensor embarcado, instrumentado com uma placa de desenvolvimento ESP32 e sensores destinados à coleta de dados de temperatura, umidade e peso, utilizando-se comunicação sem fio. Adicionalmente, foi desenvolvida uma plataforma *web* composta por um *backend* robusto e uma aplicação interativa. O armazenamento, processamento e a visualização dos dados em tempo real são realizados por esta plataforma, permitindo ao usuário o gerenciamento de múltiplos apiários, o acesso a históricos, o registro de manejos e colheitas, além do recebimento de alertas automáticos sobre condições críticas, como perdas de peso significativas. Por meio do sistema proposto, busca-se oferecer aos apicultores uma ferramenta de monitoramento que otimize o manejo das colônias, reduza a necessidade de visitas frequentes ao apiário e contribua para o aumento da produtividade, com ênfase em baixo custo, eficiência energética e escalabilidade.

Palavras-chave: Apicultura, *Apis mellifera*, Internet das Coisas, ESP32, Monitoramento, Sensores.

Abstract

The development of a comprehensive Internet of Things (IoT) system designed for the remote management and monitoring of beehives of the *Apis mellifera* species in rational beekeeping is presented in this work. An embedded sensor node, equipped with an ESP32 development board and sensors for collecting temperature, humidity, and weight data, is integrated into the solution using wireless communication. Additionally, a cloud-based *web* platform was developed, consisting of a robust *backend* and an interactive application. Data storage, processing, and real-time visualization are performed by this platform, allowing users to manage multiple apiaries, access historical data, log management activities and harvests, and receive automatic alerts regarding critical conditions, such as significant weight loss. Through the proposed system, it is aimed to provide beekeepers with a monitoring tool that optimizes colony management, reduces the need for frequent apiary visits, and contributes to increased productivity, focusing on low cost, energy efficiency, and scalability.

Keywords: Beekeeping, *Apis mellifera*, Internet of Things, ESP32, Monitoring, Sensors.

Sumário

Lista de Siglas e Abreviaturas	xii
---------------------------------------	------------

Lista de Figuras	xiii
-------------------------	-------------

1	Introdução	1
1.1	Contextualização	1
1.2	Objetivos	2
1.2.1	Objetivo geral	2
1.2.2	Objetivos específicos	2
1.3	Justificativa	3
1.4	Estrutura do Trabalho	3
2	Fundamentação Teórica	4
2.1	Abelhas e sua importância	4
2.1.1	A espécie <i>Apis mellifera</i> L. (Abelha Africanizada)	4
2.1.2	Comportamento e ciclo de vida da colônia	5
2.1.3	Introdução da <i>Apis mellifera</i> no Brasil	6
2.1.4	Espaço abelha e colmeia padrão (Langstroth)	6
2.1.5	Sazonalidade da abelha africanizada	7
2.1.6	Produtividade da abelha africanizada	8
2.1.7	Importância ecológica	9
2.1.8	Relevância econômica	9
2.1.9	Desafios da apicultura moderna	9
2.2	Internet das coisas	10
2.2.1	A IoT e o agronegócio	10
2.2.2	IoT na apicultura	10
2.2.3	Arquitetura de sistemas IoT	11
2.3	Tecnologias de comunicação para IoT	12
2.3.1	LoRa	13
2.4	Principais protocolos de rede	13
2.4.1	HTTP/HTTPS	13
2.4.2	LoRaWAN	14
2.4.3	MQTT	15
2.5	Microcontroladores	15
2.5.1	Microcontroladores ESP32	15
2.6	Sensores e tecnologias de monitoramento	16
2.6.1	Transdutores	16
2.6.2	Extensômetros	17
2.6.3	Células de carga	17

2.6.4	Conversores A/D	18
2.6.5	Firmware	19
2.6.6	Backend	19
2.6.7	Front-end	20
3	Metodologia e desenvolvimento	21
3.1	Abordagem metodológica	21
3.2	Requisitos do sistema	22
3.2.1	Definição da arquitetura do sistema	24
3.3	Desenvolvimento do Hardware	24
3.3.1	Seleção de Componentes e arquitetura do Hardware	24
3.3.2	Consumo elétrico	30
3.3.3	Implementação do módulo LoRa	31
3.4	Desenvolvimento do <i>Firmware</i>	31
3.4.1	Lógica de Execução e Gerenciamento de Energia (<i>Deep Sleep</i>)	32
3.4.2	Modos de Operação e Interface	32
3.4.3	Coleta, Formatação e <i>Timestamping</i>	33
3.4.4	Conectividade e Resiliência	33
3.5	Banco de Dados e Mensageria	35
3.5.1	Modelagem e Estrutura dos Dados	35
3.5.2	Otimização de Séries Temporais (<i>TimescaleDB</i>)	36
3.5.3	Mensageria com Eclipse Mosquitto	37
3.6	Desenvolvimento <i>backend</i> (API)	37
3.6.1	Seleção de Tecnologias	37
3.6.2	Arquitetura Modular	38
3.6.3	Comunicação e Ingestão de Dados (MQTT)	38
3.6.4	Interface de API, Segurança e Documentação	38
3.6.5	Automação e Detecção de Anomalias	39
3.6.6	Infraestrutura, <i>DevOps</i> e Implantação	39
3.7	Desenvolvimento da Interface do Usuário (<i>frontend</i>)	40
3.7.1	Seleção de Tecnologias	41
3.7.2	Arquitetura de Comunicação e Gerenciamento de Estado	41
3.7.3	Funcionalidades e Telas Principais	42
3.7.4	Experiência do Usuário e Paginação no Lado do Servidor	43
3.7.5	Infraestrutura, Hospedagem e Entrega Contínua	43
4	Resultados	44
4.1	Montagem e validação do nó sensor	44
4.1.1	Validação de Confiabilidade da balança	45
4.1.2	Validação de Confiabilidade e Estabilidade da Telemetria	47
4.1.3	Validação do Mecanismo de Tolerância a Falhas (<i>Buffer Offline</i>)	49
4.2	Apresentação da plataforma Web	51
4.2.1	Inicialização e Configuração de Conectividade	51
4.2.2	<i>Dashboard</i>	52
4.2.3	Apiários	52
4.2.4	Gerenciamento de alertas	55
4.2.5	Histórico da colmeia	58

5	Conclusão e trabalhos futuros	63
5.1	Conclusão	63
5.2	Limitações do Trabalho	63
5.3	Trabalhos futuros	64
	Referências Bibliográficas	66
	Apêndices	69
A	Links dos Repositórios e Aplicação	70
A.1	Código-Fonte (Repositórios GitHub)	70
A.2	Acesso à Aplicação	70

Lista de Siglas e Abreviaturas

A/D	Analógico para Digital (Analog to Digital)
API	Interface de Programação de Aplicação (Application Programming Interface)
BLE	Bluetooth de Baixo Consumo de Energia (Bluetooth Low Energy)
CDN	Rede de Distribuição de Conteúdo (Content Delivery Network)
CPU	Unidade Central de Processamento (Central Processing Unity)
CRUD	Criar, Ler, Atualizar e Remover (Create, Read, Update and Delete)
CSS	Espalhamento Espectral por Chirp (Chirp Spread Spectrum)
DTO	Objetos de Transferência de Dados (Data Transfer Objects)
GIS	Sistema de Informação Geográfica (Geographic Information System)
GPIO	Entrada e Saída de Uso Geral (General-Purpose Input/Output)
HTTP	Protocolo de Transferência de Hiper Texto (HyperText Transfer Protocol)
HTTPS	Protocolo de Transferência de Hiper Texto (HyperText Transfer Protocol)
IOT	Internet das Coisas (Internet Of Things)
ISM	Industrial, Científica e Médica (Industrial, Scientific and Medical)
JSON	Objeto de Notação JavaScript (JavaScript Object Notation)
JWT	Token Web JSON (JSON Web Token)
LORA	Comunicação de Longo Alcance (Long Range)
MQTT	Transporte de Telemetria por Enfileiramento de Mensagens (Message Queuing Telemetry Transp
NB-IOT	Internet das Coisas em Banda Estreita (Narrowband Internet of Things)
OTA	Sobre O Ar (Over The Air)
QOS	Qualidade de Serviço (Quality Of Service)
RAM	Memória de Acesso Randômico (Random Acess Memory)
RTC	Relógio de Tempo Real (Real Time Clock)
SDK	Kit de Desenvolvimento de Software (Software Development Kit)
TLS	Segurança da Camada de Transporte (Transport Layer Security)
WI-FI	Rede Sem Fio (Wireless Fidelity)

Lista de Figuras

2.1	Classificação das abelhas <i>Apis</i>	5
2.2	Ilustração da colmeia padrão Langstroth (Americana).	7
2.3	Parâmetros de produtividade de mel (kg) e o comportamento defensivo.	8
2.4	Ilustração da arquitetura de sistemas IoT.	11
2.5	Ilustração da arquitetura LoRaWan.	14
2.6	ESP32 WROOM.	15
2.7	Representação de um extensômetro.	17
2.8	Saída de um circuito Sample & Hold quando estimulada por um sinal contínuo.	18
3.1	Diagrama de blocos.	24
3.2	Ilustração e pinout do ESP32-S3 DevKitC-1.	25
3.3	RTC 3231.	26
3.4	Módulo HX711.	26
3.5	Célula de carga de 50 kg.	26
3.6	Módulo sensor de temperatura AHT10.	27
3.7	Módulo sensor de temperatura DS18B20.	27
3.8	Diagrama elétrico	29
3.9	Teste de consumo máximo e mínimo.	30
3.10	Representação do sistema em máquina de estados.	34
3.11	Modelagem relacional	36
3.12	Máquina Virtual (Droplet) na Digital Ocean	40
4.1	Estrutura da balança corporal.	44
4.2	Módulo HX711 acoplado na estrutura.	45
4.3	Protótipo acoplado na colmeia <i>Langstroth</i>	45
4.4	Dados armazenados no banco de dados durante o teste.	48
4.5	Dados armazenados no banco de dados durante o teste.	50
4.6	Tela <i>captive portal</i>	51
4.7	Tela de Configuração.	51
4.8	Tela de <i>dashboard</i>	52
4.9	Tela da listagem de apiários.	53
4.10	Tela da listagem de colmeias do apiário.	53
4.11	Tela da listagem de colheitas.	54
4.12	Tela de registro de colheita.	55
4.13	Notificações de alertas.	56
4.14	Tela de exibição de alertas novos.	56
4.15	Tela de exibição de alertas lidos.	57
4.16	Tela de exibição de alertas resolvidos.	58
4.17	Painel de sensoriamento da colmeia	59
4.18	Gráficos de temperatura e umidade	60

4.19	Tela de manejos da colmeia	61
4.20	Exibição dos dados no Google sheets	62

Capítulo 1

Introdução

1.1 Contextualização

A apicultura, prática milenar de manejo de abelhas para produção de mel, própolis, cera e outros derivados, tem se consolidado como atividade estratégica frente aos desafios contemporâneos do setor agropecuário. Em especial, a escassez de mão de obra qualificada e o êxodo rural têm comprometido a viabilidade de sistemas produtivos tradicionais. Segundo o Instituto Brasileiro de Geografia e Estatística - (IBGE, 2024), houve uma redução significativa da população economicamente ativa em áreas rurais nas últimas décadas, impactando diretamente atividades agropecuárias especializadas como a criação de *Apis mellifera*.

Além de seu valor econômico, a apicultura desempenha papel crucial na manutenção da biodiversidade e na polinização de culturas agrícolas, sendo considerada uma atividade de interface entre produção e conservação ambiental. A crescente demanda por produtos naturais e sustentáveis tem impulsionado o interesse por soluções tecnológicas que viabilizem a modernização da apicultura sem comprometer sua essência ecológica.

Nesse cenário, tecnologias emergentes como a Internet das Coisas *Internet of Things*(IOT), sistemas embarcados têm sido incorporadas ao manejo apícola com resultados promissores, a exemplo do Apis Online (CIRAM, 2025) e (MOJARAVSKI, 2018). O uso de microcontroladores como o ESP32 permite a integração de sensores para a coleta de dados vitais da colmeia, como peso, temperatura e umidade, e módulos de comunicação sem fio, viabilizando o monitoramento remoto e em tempo real das condições das colmeias. Essa abordagem reduz a necessidade de visitas frequentes ao apiário, o que é especialmente relevante em regiões de difícil acesso ou com pouca oferta de mão de obra.

No contexto brasileiro, iniciativas de extensão rural e pesquisa aplicada têm incentivado a adoção dessas tecnologias por pequenos e médios produtores. Projetos vinculados a instituições como a Embrapa, instituições federais e cooperativas regionais têm demonstrado que a digitalização da apicultura pode ser acessível e escalável, desde que acompanhada de capacitação técnica e políticas públicas de incentivo à inovação.

Diante do exposto, o propósito do presente trabalho consiste no desenvolvimento de um sis-

tema IoT de ponta a ponta para a gestão e monitoramento de colmeias de *Apis mellifera* utilizando a plataforma ESP32, com baixo consumo energético, integração de células de carga, temperatura e umidade, com comunicação de longo alcance via LoRaWAN e uma plataforma *web* de gestão, com o objetivo de oferecer uma solução eficiente, de baixo custo e adaptável às diferentes realidades da apicultura brasileira. A proposta se insere na interseção entre engenharia de computação e sustentabilidade produtiva, buscando contribuir para a modernização do setor, a valorização do conhecimento técnico e a promoção de uma apicultura mais resiliente e inteligente.

1.2 Objetivos

1.2.1 Objetivo geral

Desenvolver e validar um sistema de monitoramento e gerenciamento de colmeias de abelhas do gênero *Apis* baseado em tecnologias de IoT, utilizando o microcontrolador ESP32, sensores ambientais e células de carga, conectado a uma plataforma *web* para visualização de dados, de modo a auxiliar apicultores na tomada de decisão, e contribuir para a preservação das colônias e aumento de produtividade.

1.2.2 Objetivos específicos

- Projetar e construir o hardware embarcado (Nó Sensor) utilizando o microcontrolador ESP32-S3, células de carga, temperatura e umidade, com comunicação de longo alcance via LoRa;
- Desenvolver o *firmware* do Nó Sensor, incorporando estratégias de baixo consumo energético (como o modo *deep sleep*) e mecanismos de *buffer* para garantir a integridade dos dados em caso de falhas de comunicação;
- Configurar e integrar um *Gateway* LoRaWAN, capaz de receber dados de múltiplos nós sensores e retransmiti-los para a nuvem por meio de conexão Wi-Fi ou 4G;
- Implementar uma infraestrutura de *backend* escalável, utilizando *Node.js*, *Nest.js* e banco de dados *PostgreSQL* com *TimescaleDB*, para armazenamento eficiente de séries temporais e comunicação via protocolo MQTT;
- Desenvolver uma aplicação *web* responsiva, com interface intuitiva para visualização gráfica dos dados dos sensores, além de funcionalidades de gestão de apiários, colmeias, registros de manejo e colheitas;
- Criar um sistema de alertas automatizados, capaz de notificar o apicultor sobre anomalias detectadas, como variações bruscas de temperatura ou perda de peso, indicando possíveis riscos à saúde da colônia.

1.3 Justificativa

A apicultura desempenha papel fundamental na economia rural e na conservação ambiental, sendo responsável não apenas pela produção de mel e derivados, mas também pela polinização de culturas agrícolas e manutenção da biodiversidade. No entanto, o setor enfrenta desafios significativos, como a escassez de mão de obra qualificada, o envelhecimento da população rural e a dificuldade de acesso a tecnologias apropriadas, especialmente entre pequenos e médios produtores.

Nesse cenário, a aplicação de tecnologias emergentes, como sistemas embarcados e Internet das Coisas (IoT), representa uma oportunidade estratégica para modernizar o manejo apícola, tornando-o mais eficiente, sustentável e acessível. O uso de microcontroladores como o ESP32, aliado a sensores ambientais e células de carga, permite o monitoramento remoto e em tempo real das condições das colmeias, reduzindo a necessidade de visitas frequentes ao apiário e possibilitando intervenções rápidas diante de anomalias.

Além disso, a integração desses dados a uma plataforma *web* de visualização e análise oferece ao apicultor uma ferramenta poderosa de gestão, capaz de gerar alertas, consolidar históricos de produção e apoiar a tomada de decisão baseada em evidências. Essa abordagem contribui diretamente para a preservação das colônias, aumento da produtividade e redução de perdas, promovendo maior previsibilidade e segurança na atividade.

A proposta deste projeto se justifica, portanto, pela necessidade de soluções tecnológicas que atendam às demandas reais do campo, com foco na escalabilidade, baixo custo e facilidade de uso. Ao unir conhecimentos de engenharia eletrônica, ciência de dados e práticas zootécnicas, o sistema desenvolvido busca não apenas resolver problemas operacionais, mas também fomentar a inovação no setor apícola brasileiro, fortalecendo sua competitividade e sustentabilidade a longo prazo.

1.4 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos principais. O Capítulo 1 apresenta a introdução, estabelecendo a contextualização do problema, os objetivos geral e específicos, além da justificativa para o desenvolvimento da solução. O Capítulo 2 fornece a fundamentação teórica, abordando a biologia e importância da *Apis mellifera*, os conceitos de Internet das Coisas (IoT) aplicados ao agronegócio e as tecnologias de comunicação e transdução. O Capítulo 3 detalha a metodologia e o desenvolvimento técnico integral, descrevendo desde a arquitetura do *hardware* e *firmware* até a modelagem do banco de dados e a construção das camadas de *backend* e *frontend*. No Capítulo 4, são apresentados os resultados obtidos, focando na validação do nó sensor, na confiabilidade da telemetria e na demonstração das funcionalidades da plataforma *web*. Por fim, o Capítulo 5 encerra o documento com a conclusão do projeto, limitações identificadas durante o desenvolvimento e propostas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Para construir um projeto cujo o sistema é capaz de coletar e monitorar a evolução das colônias de abelhas *Apis*, foi necessário realizar estudos que abrangem o impacto da apicultura em caráter econômico e ecológico, selecionar e os periféricos a serem controlados pelo microcontrolador, tecnologias e protocolos eficientes que visam a economia de consumo elétrico e transmissão de dados para o melhor funcionamento de um sistema de gestão e monitoramento.

2.1 Abelhas e sua importância

As abelhas são agentes polinizadores essenciais para a manutenção da biodiversidade vegetal. Embora muitas pessoas as associem apenas à produção de mel, sua principal contribuição é a polinização, processo pelo qual o pólen é transportado de uma flor para outra, permitindo a reprodução de inúmeras espécies de plantas. Durante a coleta de néctar e pólen, esses insetos promovem a fecundação cruzada das plantas, garantindo a formação de frutos e sementes viáveis. De acordo com ROUBIK (1995), em estudo focado nas regiões tropicais, estima-se que cerca de 70% das culturas agrícolas dessas áreas dependem, em alguma medida, da polinização por insetos, sendo as abelhas as principais responsáveis por esse serviço ecossistêmico, essencial para a produção de alimentos no planeta. Além da polinização e da produção de mel, as abelhas fornecem substâncias valiosas, tais como cera, própolis, geleia real e apitoxina, todas amplamente utilizadas pelo ser humano como alimento, matéria-prima ou insumos medicinais. As abelhas nativas das Américas incluem milhares de espécies pertencentes a diferentes gêneros. Entre os grupos mais conhecidos estão as abelhas sem ferrão (tribo Meliponini),

2.1.1 A espécie *Apis mellifera* L. (Abelha Africanizada)

Dentre as espécies de abelhas conhecidas, a *Apis mellifera*, popularmente chamada de abelha-europeia ou abelha-ocidental, é uma espécie social que se destaca como uma das principais polinizadoras e produtoras de mel em escala global, sendo também a mais difundida e estudada. As abelhas europeias, como a *A. m. mellifera* e a *A. m. ligustica*, são conhecidas por sua docilidade e

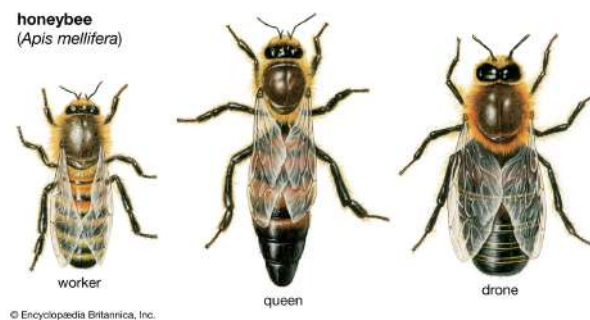
alta produtividade em climas temperados, além de apresentarem um comportamento de enxameação mais contido, isto é, a ascensão de uma nova rainha que leva parte da colônia para formar uma nova colônia, uma divisão da colônia original, que garante a perpetuação da espécie. Por outro lado, as abelhas africanas, como a *A. m. scutellata*, destacam-se por sua elevada capacidade reprodutiva, comportamento mais defensivo e excelente adaptação a ambientes quentes e com menor oferta de recursos.

A coexistência dessas diferentes subespécies, especialmente após a introdução das abelhas africanas no Brasil, desencadeou um intenso processo de hibridização que originou a chamada abelha africanizada. Essa nova linhagem híbrida, ao herdar o comportamento defensivo e a elevada capacidade de adaptação das abelhas africanas, bem como a alta produtividade das europeias, estabeleceu-se como uma das linhagens de *Apis mellifera* mais bem-sucedidas em ambientes neotropicais, exercendo um impacto profundo sobre a apicultura, a agricultura e a ecologia do país (JONG, 1996).

2.1.2 Comportamento e ciclo de vida da colônia

As colônias de *Apis mellifera* são altamente organizadas. Por serem insetos sociais, as abelhas vivem em uma estrutura social complexa, dividida em três castas: a rainha, as operárias e os zangões, conforme representado na Figura 2.1.

Figura 2.1: Classificação das abelhas *Apis*.



Fonte: ENCYCLOPEDIA BRITANNICA (2025)

O ciclo começa com a rainha, que é a única fêmea fértil da colônia e pode viver de dois a cinco anos. Sua principal tarefa é a postura de ovos, podendo chegar a colocar milhares por dia durante os períodos de alta atividade, o que garante a continuidade da colônia. Esses ovos dão origem a larvas que podem se desenvolver em operárias, zangões ou, em situações especiais, em uma nova rainha, dependendo do tipo de alimentação fornecida pelas operárias, especialmente quanto à oferta de geleia real. As operárias, que vivem em média 45 dias durante a primavera e o verão, ou até alguns meses no inverno, passam por uma divisão de trabalho ligada à idade. Quando jovens, atuam dentro da colônia como “abelhas domésticas”: limpam as células, alimentam a cria, produzem cera e constroem os favos. Em seguida, passam a atuar como guardas da colmeia, defendendo a entrada contra intrusos. Finalmente, quando atingem maior maturidade, tornam-se forrageadoras, também

chamadas de campeiras, saindo em busca de néctar, pólen, água e resinas para a produção de mel e própolis, que sustentam a colônia. Uma colônia de *Apis mellifera* pode viver durante muitos anos, mas seu equilíbrio depende da saúde e produtividade da rainha, da força das operárias e da ausência de problemas como infestações por parasitas. Assim, o ciclo de vida da colônia configura um sistema dinâmico que envolve nascimento, trabalho cooperativo, reprodução e substituição da rainha sempre que necessário, garantindo a continuidade desse superorganismo ao longo do tempo.

2.1.3 Introdução da *Apis mellifera* no Brasil

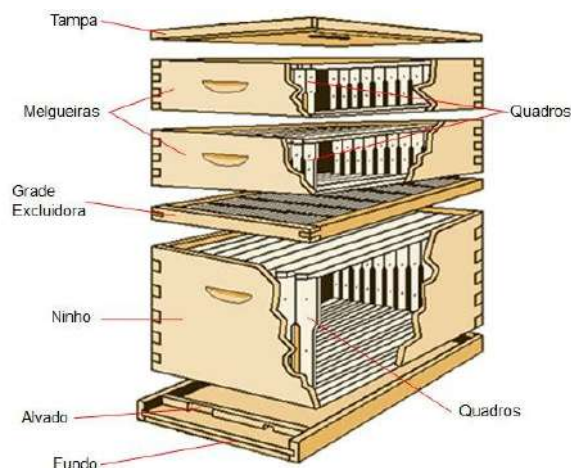
A apicultura brasileira iniciou-se no século XIX com a introdução de subespécies europeias de *Apis mellifera*, como a *mellifera*, *ligustica* e *carnica*, que apresentaram dificuldades de adaptação ao clima tropical (JONG, 1996). Em 1956, Warwick Estevam Kerr introduziu no estado de São Paulo rainhas da subespécie africana *Apis mellifera scutellata*, visando maior produtividade e adaptação. A fuga de enxames em 1957 resultou em intensa hibridização com as linhagens europeias, originando as chamadas abelhas africanizadas (JONG, 1996). Essas populações híbridas, mais resistentes e adaptáveis, rapidamente se espalharam pelo Brasil e pelo continente americano, tornando-se fundamentais para a polinização e produção de mel, além de impactarem de forma significativa a apicultura e agricultura nacional.

2.1.4 Espaço abelha e colmeia padrão (Langstroth)

A apicultura moderna, tal como é praticada no Brasil e na maior parte do mundo, caracteriza-se pelo uso da colmeia de quadros móveis, conhecida como Colmeia Padrão *Langstroth* ou Colmeia Americana. Desenvolvida pelo apicultor e pastor Lorenzo Lorraine Langstroth nos Estados Unidos em 1851, essa invenção revolucionou a atividade ao permitir o manejo e a inspeção das colônias sem destruir o ninho, prática antes comum em colmeias rústicas.

O sucesso do modelo Langstroth reside na aplicação do conceito de espaço-abelha (*bee space*). Langstroth observou que as abelhas mantêm, de forma consistente, um espaço livre de 6mm a 9,5mm entre os favos e as paredes da colmeia. Quando o espaço é menor que 6,5mm, as abelhas o preenchem com própolis; quando é maior que 9,5mm, elas o ocupam com cera, construindo favos indesejados (APACAME, 2022).

Figura 2.2: Ilustração da colmeia padrão Langstroth (Americana).



Fonte: MICHELS (2016)

Cada componente da colmeia apresentado na Figura 2.2 tem uma função específica, sendo:

1. Tampa: protege o interior da colmeia contra o frio e a entrada de elementos prejudiciais.
2. Melgueiras: parte com quadros onde é depositado o mel.
3. Quadros: são armações onde as abelhas vão construir os favos de cera com alvéolos hexagonais, para criar filhotes e deposição de mel, água e pólen.
4. Grade Excludora: é colocada para evitar que a rainha acesse as melgueiras destinadas à produção de mel.
5. Ninho: parte com quadros destinados a depósito de mel, pólen ou crias.
6. Alvado: a porta da colmeia, por onde ocorre a entrada e saída das abelhas.
7. Fundo: parte inferior da colmeia.

Ao projetar seus quadros de modo que todas as distâncias entre eles e as paredes da caixa, bem como entre os quadros superiores e inferiores, respeitassem precisamente esse intervalo, Langstroth garantiu que os quadros pudessem ser removidos com facilidade. O quadro móvel, fundamental para o manejo sanitário, a colheita do mel e a inspeção da rainha, constitui a base para uma gestão mais eficiente das colmeias, permitindo a instalação de sensores em locais específicos e padronizados para o monitoramento eletrônico.

2.1.5 Sazonalidade da abelha africanizada

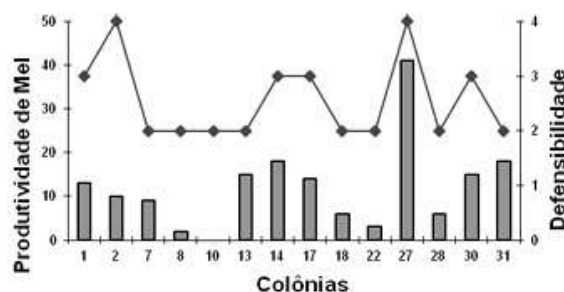
A sazonalidade em abelhas africanizadas, entendida como variações comportamentais e produtivas ao longo do ano, influencia significativamente aspectos da colônia, como defensividade, enxameação e produção de mel. Defensibilidade: Conforme relatado por SILVEIRA et al. (2024),

na região do sertão paraibano, durante a estação chuvosa (abril), as colmeias apresentam maior defensividade, caracterizada por menor tempo de resposta ao distúrbio e perseguições mais prolongadas, especialmente entre 15h e 17h, devido ao aumento populacional e ao retorno das campeiras. No período seco (agosto), observa-se redução da agressividade, associada à menor densidade populacional e à escassez de recursos, tornando as colônias mais vulneráveis. Enxameação: Durante as chuvas e floradas, há crescimento populacional e picos de enxameação. Já na seca, ocorre redução da postura e dos estoques, com menor atividade forrageira. No período de seca, a atividade externa diminui, exigindo suplementação alimentar para evitar a perda do enxame. A produção acompanha o padrão sazonal. Nos períodos de chuva e floração, há maior coleta de néctar, resultando em aumento da população e da defensividade. Em contraste, na seca, a escassez de recursos limita a produção e altera o comportamento da colônia.

2.1.6 Produtividade da abelha africanizada

Segundo LIMA (2012), a produtividade da abelha africanizada (*Apis mellifera* L.), em termos de produção de mel, é de cerca de 40 kg por colmeia por ano. Com manejo adequado e condições ambientais favoráveis, essa produtividade pode ser elevada para aproximadamente 60 kg por colmeia. Além de apresentar boa produtividade de mel, a abelha africanizada é adaptada ao clima tropical, apresenta alta resistência a doenças e tem facilidade para enxamear, o que favorece sua utilização na apicultura comercial. Além disso, pesquisas recentes da EMBRAPA (2024) mostram que a inserção de abelhas africanizadas em culturas agrícolas, como cafezais e lavouras de canola (ANTUNES, 2024), pode aumentar a produtividade das plantações em valores entre 16,5% e 30%, refletindo um benefício econômico significativo para os agricultores. Um estudo realizado por pesquisadores da UFBA (CASTAGNINO, 2012), que relacionou a defensibilidade das abelhas africanizadas com a produtividade de mel, constatou que a produtividade é diretamente proporcional à defensibilidade das colônias, conforme é representado na Figura 2.3.

Figura 2.3: Parâmetros de produtividade de mel (kg) e o comportamento defensivo.



Fonte: CASTAGNINO (2012)

2.1.7 Importância ecológica

A apicultura transcende a mera produção de mel e demais produtos apícolas, assumindo um papel de serviço ecossistêmico fundamental. A gestão de colmeias ativas e sadias pelos apicultores não apenas mantém, mas intensifica a polinização cruzada, processo biológico crucial para a reprodução sexuada de vastas populações de angiospermas, tanto em sistemas agrícolas quanto em ambientes naturais. Esse serviço é vital, pois assegura a produtividade das culturas e, consequentemente, contribui para a segurança alimentar, ao mesmo tempo em que atua como pilar na conservação da biodiversidade vegetal, visto que a visita das abelhas é um vetor reprodutivo indispensável para inúmeras espécies. Adicionalmente, a atividade apícola atua como um mecanismo de mitigação da degradação ambiental. A necessidade de recursos forrageiros para as abelhas incentiva a restauração de *habitats*, por meio do plantio e da conservação de espécies nativas e floríferas, o que indiretamente promove a melhoria da qualidade do solo e dos recursos hídricos. Um aspecto crucial é o papel das abelhas como bioindicadores ambientais. O monitoramento constante da saúde das colmeias pelos apicultores possibilita a detecção precoce de estressores ambientais, como a contaminação por agrotóxicos.

2.1.8 Relevância econômica

A apicultura possui grande importância econômica, gerando renda, diversificando a produção agrícola e sustentando cadeias produtivas. Além do mel, produtos como cera, própolis, geleia real e pólen apresentam alto valor agregado e aplicações em setores alimentícios, cosméticos e farmacêuticos (JONG, 1996).

Outro fator essencial é a polinização, que aumenta a produtividade de culturas como café e hortaliças, fortalecendo a agricultura e o comércio. De acordo com o MAPA (2021), em 2020 o Brasil exportou 45,7 mil toneladas de mel natural, cerca de 50% a mais que em 2019, tendo os Estados Unidos como principal destino. Apesar do crescimento, o país ainda possui elevado potencial de expansão, especialmente com práticas de manejo aprimoradas e integração a culturas dependentes da polinização.

2.1.9 Desafios da apicultura moderna

Apesar de sua resiliência e adaptabilidade, as populações de *Apis mellifera* e, consequentemente, a apicultura moderna enfrentam desafios crescentes que comprometem a produção e a sustentabilidade da atividade. Um dos principais problemas é o uso indiscriminado de agrotóxicos na agricultura, que afeta diretamente a saúde das abelhas e de outros insetos, causando desorientação, enfraquecimento do sistema imunológico e até morte em larga escala. Somam-se a esses fatores a proliferação de doenças e parasitas, como o ácaro *Varroa destructor*, que podem enfraquecer colônias inteiras. A apicultura também enfrenta um desafio significativo no que diz respeito à mão de obra. A escassez de mão de obra no campo brasileiro tem se intensificado, impulsionada pelo êxodo rural, pela urbanização acelerada e pelo envelhecimento da população rural. Segundo MA-

LAFIAIA (2024), o número de trabalhadores no setor primário caiu de 10,23 milhões em 2012 para 8,25 milhões em 2023, uma redução de 19,35% em uma década. Esse fenômeno, frequentemente denominado “apagão de mão de obra”, compromete a sustentabilidade e a eficiência das atividades agropecuárias, incluindo a apicultura. GAZZONI (2018) destaca que os desafios são tanto quantitativos quanto qualitativos, em razão da crescente demanda por profissionais com competências técnicas para operar sistemas automatizados. Esse cenário exige soluções tecnológicas que reduzam a dependência de mão de obra humana e promovam maior eficiência operacional. A automação e a digitalização das atividades rurais surgem como alternativas viáveis para enfrentar esse desafio, especialmente em setores mais especializados, como a apicultura.

2.2 Internet das coisas

A Internet das Coisas (IoT) consolida-se como um paradigma tecnológico que expande as fronteiras da conectividade global. Conforme definem Santos et al. (2016), “a Internet das Coisas, em poucas palavras, nada mais é que uma extensão da Internet atual, que proporciona aos objetos do dia-a-dia (quaisquer que sejam), mas com capacidade computacional e de comunicação, se conectarem à Internet” (SANTOS et al., 2016). Essa conexão viabiliza não apenas o controle remoto, mas tem o objetivo de permitir que os próprios objetos sejam acessados como provedores de serviços. Para sustentar esse ecossistema, é fundamental a presença dos chamados objetos inteligentes.

2.2.1 A IoT e o agronegócio

No setor agropecuário, por exemplo, a IoT tem se tornado essencial para enfrentar desafios como a escassez de mão de obra qualificada, bem como para otimizar o manejo de recursos e impulsionar a produtividade por meio da agricultura e da pecuária de precisão. A IoT tem ganhado grande relevância no agronegócio, sendo incorporada em diversos projetos de pesquisa e em iniciativas da iniciativa privada, oferecendo soluções inovadoras que otimizam a produção, aumentam a rastreabilidade, reduzem custos e promovem a sustentabilidade em todas as etapas da cadeia produtiva. No Brasil, essa tecnologia apresenta grande potencial de aplicação no campo, especialmente em atividades que demandam monitoramento contínuo e controle remoto, como a apicultura. Nesse contexto, a aplicação da IoT na apicultura moderna representa um avanço significativo. Ao permitir o monitoramento remoto e contínuo de parâmetros vitais das colmeias, como peso, temperatura e umidade, a tecnologia transforma o manejo tradicional em um processo mais preciso, eficiente e orientado por dados.

2.2.2 IoT na apicultura

As aplicações da IoT no campo são vastas e abrangem desde o monitoramento de culturas e rebanhos até a gestão de equipamentos e recursos hídricos. Na apicultura, sistemas embarcados

têm sido utilizados para integrar sensores de temperatura, umidade, luminosidade e peso, permitindo o acompanhamento em tempo real das condições das colmeias. (MACHADO, 2023) aponta que a adoção de tecnologias de precisão no manejo apícola contribui para a redução de perdas por abandono de colônias, melhora a rastreabilidade da produção e facilita a tomada de decisão baseada em dados. Além disso, a IoT favorece a transparência na cadeia produtiva, atributo valorizado em mercados internacionais. A revisão sistemática publicada na Revista Interações (2024) (KUBOTA, 2023) reforça que a aplicação da IoT na agropecuária brasileira está diretamente relacionada à modernização das práticas produtivas, à sustentabilidade operacional e à superação de limitações estruturais, como a escassez de mão de obra qualificada. Dessa forma, a utilização da IoT na apicultura representa uma convergência entre inovação tecnológica e sustentabilidade produtiva, oferecendo soluções práticas e escaláveis para os desafios enfrentados pelo setor. O uso de microcontroladores como o ESP32, aliado a sensores ambientais e plataformas de análise, configura um avanço significativo na gestão inteligente das colmeias. Assim, o apicultor consegue utilizar dados concretos para tomar decisões mais embasadas, otimizando a saúde e a produtividade de suas abelhas e de todo o processo produtivo.

2.2.3 Arquitetura de sistemas IoT

A arquitetura de sistemas IoT é fundamental para a viabilização da comunicação entre objetos físicos e o mundo digital. Segundo SANTOS et al. (2016), o modelo básico dessa arquitetura organiza-se em quatro camadas principais: a camada de percepção, que representa os dispositivos físicos responsáveis pela coleta e processamento de informações; a camada de rede, onde ocorrem as abstrações de comunicação, roteamento e identificação; a camada de processamento, que consiste no armazenamento e disponibilização dos dados coletados; e, por fim, a camada de aplicação, a qual é responsável por prover serviços para os clientes, isto é interfaces e serviços que fornecem utilidade aos usuários finais, como aplicativos móveis, *dashboards*, sistemas de automação ou controle remoto, conforme exposto na Figura 2.4. Para suportar essa estrutura, a arquitetura de hardware dos dispositivos deve apresentar, minimamente unidade(s) de, processamento, memória, comunicação e, sensor(es) ou atuador(es).

Figura 2.4: Ilustração da arquitetura de sistemas IoT.



Fonte: Elaborado pelos autores (2025)

Essa organização evidencia que a IoT não é apenas a conexão de objetos, mas um ecossistema que depende da integração entre hardware, rede e aplicações para gerar valor real.

2.3 Tecnologias de comunicação para IoT

As tecnologias de comunicação para a Internet das Coisas (IoT) englobam diversas soluções que possibilitam a conexão e a transmissão de dados entre dispositivos inteligentes e sistemas. As principais incluem:

- LoRa (Long Range): tecnologia de comunicação sem fio de longo alcance e baixo consumo energético, ideal para aplicações rurais e urbanas com baixa taxa de dados.
- Wi-Fi: rede sem fio de alta velocidade, adequada para ambientes com energia abundante e necessidade de transmissão rápida.
- Bluetooth/Bluetooth Low Energy (BLE): tecnologia sem fio para conexões de curta distância, muito usada em dispositivos vestíveis e sensores.
- ZigBee: protocolo de rede *mesh* para comunicação de curto alcance, com foco em automação residencial e industrial.
- NB-IoT (Narrowband IoT) e LTE-M: tecnologias celulares de baixo consumo desenvolvidas para IoT, suportando ampla cobertura e conectividade de longo alcance.
- 5G: tecnologia móvel avançada que oferece alta velocidade, baixa latência e suporte massivo a dispositivos conectados, ampliando o potencial da IoT.

Essas tecnologias são escolhidas conforme os requisitos de alcance, consumo de energia, taxa de dados, ambiente de aplicação e compatibilidade com o ESP32-S3, formando a base para sistemas inteligentes em setores como agricultura, saúde, indústria e cidades inteligentes. A Tabela 2.1 compara essas tecnologias considerando os parâmetros mais importantes para a IoT.

Tabela 2.1: *Comparativo de tecnologias.*

Tecnologia	Consumo (mA)	Velocidade (bps)	Pacote (bytes)	Custo	Alcance estimado
LoRa	30 - 50	0,3 - 50 k	50 - 200	baixo	10 km
Wi-Fi	80 - 300	10 - 150 M	até 1500	(integrado)	100 m
BLE	5 - 15	1 - 2 M	20 - 244	(integrado)	100 m
ZigBee	20 - 40	20 - 250 k	80 - 100	baixo	100 m
NB-IoT	100 - 200	20 - 200 k	até 1600	médio	10 km
5G	100 - 300	100 M - 10 G	1500	alto	10 km

Alcances variam conforme potência, antena, obstáculos, frequência e regulamentação local. Valores são faixas típicas em cenários comuns. Custo relativo estimado com base em faixas de preços de módulos disponíveis no mercado de varejo (plataformas de comércio eletrônico) em 2025.

Fonte: Elaborado pelos autores (2025)

2.3.1 LoRa

O desenvolvimento e a expansão da Internet das Coisas (*IoT*) impulsionaram a necessidade de redes de comunicação capazes de conciliar longo alcance e baixo consumo de energia, superando as limitações do Wi-Fi (curto alcance e alto consumo) e das redes celulares tradicionais (alto consumo e maior custo). Nesse contexto, a tecnologia LoRa e o protocolo de rede associado, LoRaWAN, emergiram como soluções-chave, sendo amplamente adotadas em aplicações de cidades inteligentes, agricultura de precisão e monitoramento ambiental.

LoRa (acrônimo de *Long Range*) corresponde à camada física da comunicação, consistindo em uma modulação proprietária. A tecnologia foi desenvolvida inicialmente pela empresa francesa Cycleo, com suas bases estabelecidas em 2009 e posteriormente patenteada em 2014. LoRa opera predominantemente em faixas de frequência não licenciadas da banda (*Industrial, Scientific and Medical* (ISM)) em sub-GHz, tais como 433 MHz, 868 MHz e 915 MHz, variando conforme a regulamentação de cada região.

A principal característica da camada física LoRa é a utilização da modulação por *Chirp Spread Spectrum* (CSS). O CSS é uma técnica de espectro espalhado que codifica os dados por meio de pulsos de banda larga cuja frequência varia linearmente ao longo do tempo (os *chirps*). Essa modulação confere à tecnologia LoRa duas vantagens críticas:

- **Alta sensibilidade de recepção:** sinais modulados em CSS podem ser decodificados com sucesso mesmo quando sua potência está abaixo do piso de ruído (*noise floor*) do receptor. Essa característica permite alcançar orçamentos de enlace (*link budget*) da ordem de até 157 dB, possibilitando, em condições favoráveis, alcances típicos de até aproximadamente 15 km em áreas rurais e 5 km em ambientes urbanos densos.
- **Imunidade a interferências e efeitos de multipercurso:** a dispersão do espectro torna o sinal mais robusto frente a interferências de banda estreita e ao desvanecimento seletivo em frequência (*fading*) decorrente de efeitos de *multipath*.

2.4 Principais protocolos de rede

2.4.1 HTTP/HTTPS

HyperText Transfer Protocol (HTTP), ou Protocolo de Transferência de Hipertexto) é um protocolo de comunicação usado para transferir dados entre clientes (como navegadores) e servidores na internet. Ele permite que navegadores solicitem e exibam páginas da web, fazendo requisições de informações que os servidores respondem com os dados desejados (MOZILLA DEVELOPER NETWORK, 2025). Funciona no modelo cliente-servidor, onde o cliente envia uma requisição HTTP com um método (como *GET*, *POST*, *PUT*, *DELETE*) especificando a ação desejada, e o servidor responde com o recurso solicitado ou uma mensagem de status. É a base da comunicação na internet, garantindo a transferência eficiente e padronizada de hipertextos e outros conteúdos.

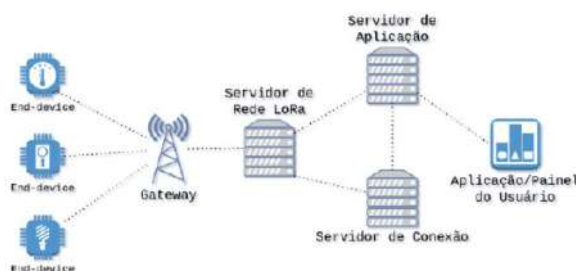
O *Hypertext Transfer Protocol Secure* (HTTPS), ou Protocolo de Transferência de Hipertexto Seguro é a versão segura do HTTP, na qual a comunicação entre cliente e servidor é protegida por uma camada de criptografia baseada, predominantemente, no protocolo *Transport Layer Security* (TLS) (GOOGLE DEVELOPERS, 2020). Em HTTPS, os dados trafegam dentro de um túnel criptografado estabelecido durante o *handshake* TLS, processo no qual o cliente e o servidor negociam algoritmos criptográficos e chaves de sessão. Além da confidencialidade, o uso de certificados digitais, emitidos por autoridades certificadoras (CAs), permite autenticar o servidor, reduzindo o risco de ataques do tipo *man-in-the-middle* (CLOUDFLARE, 2025). Comparativamente ao HTTP, que transmite os dados em texto claro, o HTTPS oferece três propriedades de segurança fundamentais: confidencialidade, integridade e autenticidade, tais características o tornam essencial para a proteção de informações sensíveis, como dados pessoais, senhas e informações financeiras. No contexto de IoT, o uso de HTTPS em APIs e serviços em nuvem aumenta a segurança das comunicações, embora introduza maior sobrecarga de processamento e consumo de energia, o que pode ser um fator limitante em dispositivos com recursos computacionais restritos, quando comparado a protocolos mais leves como *Message Queuing Telemetry Transport* (MQTT).

2.4.2 LoRaWAN

Long Range Wide Area Network (LoRaWAN) é uma especificação de camada de enlace e de rede para comunicação de longa distância com baixo consumo de energia (*Low-Power Wide-Area Networks* – LPWAN), amplamente utilizada em aplicações de Internet das Coisas (IoT) de grande escala. O protocolo é definido e mantido pela *LoRa Alliance* e opera sobre a camada física LoRa, que utiliza modulação em espectro espalhado em bandas de frequência não licenciadas na faixa sub-GHz (por exemplo, 433 MHz, 868 MHz e 915 MHz, dependendo da regulamentação regional).

A arquitetura do LoRaWAN adota uma topologia em “estrela de estrelas” composta por três elementos principais: nós finais (sensores, atuadores), *gateways* e servidores em nuvem. Os dispositivos finais comunicam-se via rádio LoRa diretamente com um ou mais *gateways*, que atuam como pontes transparentes, encaminhando os pacotes recebidos para um *network server* por meio de redes IP. A Figura 2.5 ilustra o fluxo da arquitetura do LoRaWAN.

Figura 2.5: Ilustração da arquitetura LoRaWAN.



Fonte: ERDMANN et al. (2023)

2.4.3 MQTT

MQTT (*Message Queuing Telemetry Transport*) é um protocolo de comunicação leve, padronizado pela OASIS e pela ISO/IEC, amplamente empregado na IoT. Baseia-se no modelo cliente/servidor, em que os dispositivos se conectam a um broker para publicar e assinar mensagens. Utiliza o paradigma *publish/subscribe* (editores/assinantes), no qual clientes enviam mensagens a tópicos específicos e outros, previamente inscritos, recebem essas informações.

Projetado para operar com baixo consumo de banda e recursos, o MQTT é adequado para comunicação máquina a máquina (M2M) em redes com restrições, garantindo eficiência mesmo em cenários de alta latência ou largura de banda limitada. É amplamente aplicado na integração de sensores, atuadores e dispositivos IoT, oferecendo comunicação confiável, suporte a diferentes níveis de Qualidade de Serviço (*Quality Of Service* (QOS)) e conexões persistentes entre clientes e *broker*.

2.5 Microcontroladores

Microcontroladores são circuitos integrados que combinam *Central Processing Unity* (CPU), memória não volátil (*Flash*), *Random Access Memory* (RAM), interfaces de entrada/saída (*General-Purpose Input/Output* (GPIO)) e periféricos em um único chip, possibilitando soluções de baixo custo, baixa potência e reduzida área para sistemas embarcados. Sua operação é determinada por *firmware* residente em memória não volátil, usualmente em C/C++, apoiado por interrupções e, quando necessário, por um sistema operacional de tempo real para previsibilidade temporal.

2.5.1 Microcontroladores ESP32

O microcontrolador ESP32 representado pela Figura 2.6, desenvolvido pela empresa chinesa Espressif Systems, integra conectividade sem fio (Wi-Fi e Bluetooth/BLE) e uma ampla gama de periféricos, oferecendo desempenho adequado para aplicações de IoT, automação e dispositivos conectados.

Figura 2.6: ESP32 WROOM.



Fonte: ARDUCORE (2025)

O ESP32 se destaca pela combinação de baixo consumo de energia, com suporte a modos de suspensão profunda (*deep sleep*), aceleração criptográfica em hardware e um ecossistema maduro de SDKs (*Software Development Kit* (SDK)) e bibliotecas, além de sua integração com o ecossistema Arduino, o que facilita o desenvolvimento (ESPRESSIF SYSTEMS, 2024). Tais características tornam o ESP32 particularmente adequado ao contexto deste trabalho, uma vez que oferece capacidade de processamento e conectividade sem fio suficientes para a implementação do sistema de monitoramento de colmeias proposto.

Microcontrolador ESP32-S3

O ESP32-S3 representa uma evolução estratégica na família de microcontroladores da Espressif. Ele mantém os pilares de sucesso da versão padrão do ESP32, como a conectividade sem fio (Wi-Fi e Bluetooth/BLE) e o baixo consumo, mas introduz upgrades de hardware, como memória não volátil significativamente maior (16 MB) e uma maior quantidade de GPIOs, além de seu preço acessível, que o tornam ideal para projetos mais exigentes no contexto de IoT.

2.6 Sensores e tecnologias de monitoramento

Sensores e tecnologias de monitoramento constituem elementos centrais em sistemas embarcados, aplicações de Internet das Coisas (IoT), automação industrial e diversos contextos do cotidiano. Um sensor pode ser definido como um dispositivo capaz de detectar ou medir uma grandeza física, química ou biológica — por exemplo, temperatura, pressão, iluminância, movimento, umidade, concentração de gases ou vibração — e convertê-la em um sinal elétrico ou digital passível de interpretação por microcontroladores, computadores ou sistemas supervisórios. A partir dos sinais adquiridos, procedimentos de processamento e análise permitem extrair informações relevantes, apoiar a tomada de decisão e gerar alertas ou ações de controle.

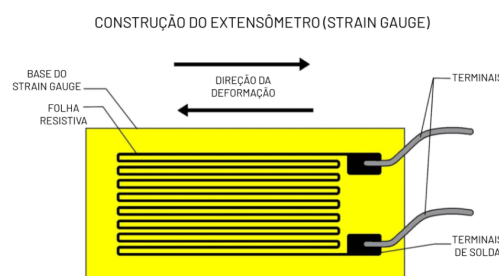
2.6.1 Transdutores

Um transdutor é um dispositivo responsável por converter uma forma de energia em outra, geralmente transformando grandezas físicas — como pressão, temperatura, força, luz ou movimento — em sinais elétricos mensuráveis, ou realizando a conversão inversa (por exemplo, de sinais elétricos em deslocamento mecânico ou luminosidade). Os transdutores são fundamentais em instrumentação, automação e sistemas de medição, pois estabelecem a interface entre fenômenos físicos e equipamentos eletrônicos. Sensores são considerados transdutores ou um conjunto de transdutores a depender do caso, pois realizam a conversão de grandezas do ambiente em sinais que podem ser processados por microcontroladores, computadores ou sistemas supervisórios.

2.6.2 Extensômetros

Extensômetro, também conhecido como *Strain gauge* é um elemento transdutor utilizado para mensurar a deformação de um material quando ele é submetido a uma força ou tensão. Ele funciona com base na propriedade de materiais condutores em que a resistência elétrica varia proporcionalmente à deformação sofrida. Ao ser colado na superfície de um material, qualquer deformação nele causa uma alteração na resistência do extensômetro, que pode ser medida eletricamente e convertida em valores de deformação, sob tração, a resistência aumenta, e sob compressão, diminui, conforme representado pela Figura 2.7.

Figura 2.7: Representação de um extensômetro.



Fonte: MAKERHERO (2025)

Este transdutor é amplamente utilizado na engenharia para testes de materiais, monitoramento estrutural e análise de componentes mecânicos, como chassis de veículos ou pontes. Geralmente, o extensômetro é conectado a um circuito de ponte de *Wheatstone* para medir com precisão as pequenas variações de resistência resultantes da deformação. Existem diferentes tecnologias de extensômetros, incluindo extensômetros de resistência elétrica metálica, semicondutores e ópticos, selecionadas conforme a faixa de medição, sensibilidade, ambiente de operação e requisitos específicos de cada aplicação.

A leitura é usualmente realizada em configuração de ponte de *Wheatstone*, possibilitando sensibilidade elevada (visto que a variação na resistência de um único extensômetro é muito baixa, na ordem de mili-ohms), rejeição de ruído e compensação térmica (meia-ponte ou ponte completa com múltiplos elementos). A sensibilidade é caracterizada pelo fator *gauge*, que relaciona a variação relativa de resistência à deformação. Além das células de carga, extensômetros são aplicados em sensores de pressão, torquímetros, acelerômetros e instrumentação experimental para análise de tensões em estruturas e componentes.

2.6.3 Células de carga

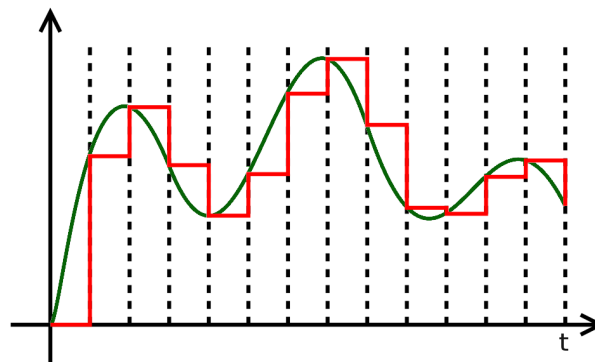
Células de carga são transdutores que convertem forças mecânicas (peso, tração ou compressão) em sinais elétricos proporcionais. A implementação mais comum emprega extensômetros afixados a um corpo metálico projetado para deformações mínimas e previsíveis; a variação de resistência sob deformação é tipicamente lida por meio de uma ponte de *Wheatstone*, resultando

em uma tensão diferencial de pequena amplitude (LOCKHAT, 2020). Devido à baixa magnitude dos sinais, utilizam-se amplificadores de instrumentação de baixo ruído e etapas de filtragem, bem como procedimentos de calibração, para vincular a tensão medida à grandeza física (força ou massa) (OMEGA ENGINEERING, 2021). Diferentes geometrias e faixas de medição são adotadas conforme a aplicação, como vigas (balanças de plataforma), compressão, tração e ponto único (balanças de bancada). A robustez, precisão e confiabilidade tornam as células de carga essenciais em setores industriais e em aplicações de IoT, como monitoramento de peso em tubulações ou silos.

2.6.4 Conversores A/D

Um conversor analógico-digital (ou ADC, do inglês *Analog-to-Digital Converter*) é um dispositivo eletrônico que transforma sinais analógicos contínuos, como tensão ou corrente elétrica, em representações digitais discretas que podem ser processadas por sistemas digitais, como microcontroladores ou computadores. Ele realiza essa conversão por meio de etapas como amostragem do sinal em intervalos periódicos e quantização, que atribui valores digitais aproximados à amplitude do sinal analógico em cada ponto de amostragem, conforme representação da Figura 2.8.

Figura 2.8: Saída de um circuito *Sample & Hold* quando estimulada por um sinal contínuo.



Fonte: EMBARCADOS (2015)

Os ADCs são essenciais em sistemas eletrônicos modernos para capturar dados do mundo real (temperatura, pressão, som, etc.) e torná-los acessíveis para processamento digital. Eles variam em resolução, velocidade e tecnologia, influenciando precisão e aplicação. O conversor ADC cria uma ponte entre o domínio analógico e digital, possibilitando a interação de dispositivos digitais com sinais do ambiente físico.

A resolução, em bits, define o número de níveis representáveis (por exemplo, $2^{10} = 1024$ para 10 bits e $2^{12} = 4096$ para 12 bits), enquanto a taxa de amostragem condiciona a faixa de frequências observável sem *aliasing*, a resolução é diretamente proporcional ao tempo de processamento.

2.6.5 Firmware

Firmware é um tipo de software de baixo nível que reside permanentemente na memória não volátil (geralmente Flash) de um dispositivo. Ele é o código fundamental que inicializa, controla e gerencia as operações de hardware do microcontrolador ou processador, permitindo que o dispositivo execute suas funções específicas e se comunique com outros componentes ou a rede (FORTINET, 2024). O *Firmware* determina a confiabilidade, consumo de energia e funcionalidade do dispositivo. O aspecto mais importante em IoT é a capacidade de realizar Atualizações *Over The Air* (OTA). Isso permite que os dispositivos sejam corrigidos contra vulnerabilidades de segurança e recebam novas funcionalidades remotamente, garantindo sua integridade e segurança no campo (ESPRESSIF SYSTEMS., 2024).

2.6.6 Backend

O *backend* constitui a camada fundamental e invisível de qualquer plataforma de gerenciamento de IoT, sendo o responsável por orquestrar a comunicação, processar os dados e gerenciar a lógica de negócio de toda a solução. Ele é a infraestrutura de software e hardware (tipicamente hospedada em nuvem) que interliga os dispositivos de borda com a interface de usuário (*frontend*), garantindo a funcionalidade, segurança e escalabilidade do sistema.

A função primária do *backend* é a Ingestão de Dados, que envolve receber e processar o fluxo contínuo e massivo de telemetria gerado por potencialmente milhões de dispositivos. Para lidar com essa escala, o *backend* utiliza *brokers* de mensagens e protocolos leves, como o MQTT, que são otimizados para redes com largura de banda limitada e conexões intermitentes. Uma arquitetura robusta de *backend* deve ser projetada com alta Escalabilidade e Resiliência, geralmente através de serviços de computação em nuvem que podem escalar dinamicamente para lidar com picos de tráfego.

Containerização

A containerização, difundida pelo *Docker*, constitui uma tecnologia que transformou o desenvolvimento e a implantação de aplicações. Ela permite empacotar programas e suas dependências em um contêiner isolado, leve e portátil, garantindo execução consistente em diferentes ambientes.

Ao encapsular bibliotecas, configurações e arquivos necessários, os contêineres solucionam o problema da incompatibilidade entre sistemas, assegurando uniformidade operacional. Diferentemente das máquinas virtuais, que replicam um sistema completo, os contêineres compartilham o *kernel* do *host*, o que reduz consumo de recursos e acelera inicialização, possibilitando maior densidade de execução em servidores.

O *Docker*, principal ferramenta da área, utiliza imagens como modelos imutáveis para criação de contêineres e oferece recursos como *Docker Compose* (gerenciamento de múltiplas aplicações) e *Docker Swarm* (orquestração em clusters).

2.6.7 Front-end

No contexto de sistemas de Internet das Coisas (IoT), a eficácia da solução não reside apenas na coleta robusta de dados pelos dispositivos de borda (*edge*) e no seu processamento eficiente no *backend* (nuvem ou servidores). A etapa final e crucial para o valor de negócio é a interação humana com esses dados, que é mediada pelo *frontnd* da plataforma de gerenciamento. O *front-end* atua como a interface visual através da qual os operadores e tomadores de decisão monitoram, controlam e gerenciam a vasta infraestrutura de dispositivos conectados. O principal objetivo do *front-end* é transformar o volume e a complexidade dos dados de IoT em informação acionável. Isso é alcançado por meio da criação de *dashboards* dinâmicos, que utilizam técnicas de Visualização de Dados para apresentar métricas de sensores (como temperatura, consumo energético, ou status de máquinas) em tempo real por meio de gráficos, *widgets* e mapas geolocalizados. Essa visualização imediata é vital para o Monitoramento de Status e Saúde dos Dispositivos, permitindo a detecção rápida de anomalias, falhas de comunicação ou alertas de segurança.

Capítulo 3

Metodologia e desenvolvimento

É detalha no capítulo 3 a metodologia de desenvolvimento adotada e a construção de todos os componentes que integram o Sistema de Gestão e Monitoramento de Colmeias. A abordagem foi estruturada em pilares que representam as principais frentes de trabalho do projeto: o *hardware* embarcado, o *firmware*, a infraestrutura de nuvem (banco de dados, *broker* MQTT e *backend*) e a interface do usuário (*frontend*).

3.1 Abordagem metodológica

Para o desenvolvimento deste projeto, que se caracteriza como uma pesquisa aplicada e um desenvolvimento tecnológico, foi adotada uma metodologia de Prototipagem Evolucionária baseada em Pilares. Esta abordagem se mostrou ideal para um sistema complexo de ponta a ponta, permitindo que a equipe fosse capaz de desenvolver, testar e validar os componentes de forma independente antes de sua integração final.

O desenvolvimento seguiu uma ordem cronológica lógica, partindo da "ponta" (a fonte de dados no campo) em direção à "nuvem" (o consumo desses dados pelo usuário). Os pilares de desenvolvimento foram definidos da seguinte forma:

- **Pilar 1: Nó Sensor (*Hardware* e *Firmware*):** O primeiro pilar focou na construção do dispositivo físico responsável pela coleta de dados. Esta etapa envolveu a seleção e montagem dos componentes eletrônicos (*Hardware*) e o desenvolvimento do *software* embarcado (*Firmware*) capaz de ler os sensores, gerenciar o consumo de energia e garantir a transmissão ou armazenamento dos dados.
- **Pilar 2: Plataforma de Nuvem (*Backend*, *broker* MQTT e Banco de Dados):** Com a fonte de dados definida, o segundo pilar concentrou-se em como recebê-los, armazená-los e gerenciá-los. Esta etapa iniciou-se pela modelagem do banco de dados, otimizado para séries temporais, e seguiu para a construção de uma Interface de Programa de Aplicação (*Application Programming Interface* — *API*) (*Backend*) responsável por intermediar a comunicação entre o banco de dados, o *broker* MQTT e a interface do usuário.

- **Pilar 3: Interface do Usuário (*Frontend*):** O último pilar foi o desenvolvimento da aplicação *web*, a camada de visualização do sistema. Esta etapa focou em traduzir os dados armazenados em informações visuais e funcionais para o apicultor, permitindo o monitoramento e a gestão dos seus ativos.

3.2 Requisitos do sistema

O levantamento de requisitos foi uma etapa fundamental para delimitar o escopo do projeto, baseando-se nos objetivos definidos. Os requisitos foram classificados em Funcionais (RF), que descrevem o que o sistema deve fazer, e Não Funcionais (RNF), conformes expostos nas Tabelas 3.1 e 3.2 que descrevem as qualidades e restrições do sistema, com base em parte da realidade da atividade apícola.

Tabela 3.1: *Requisitos Funcionais do Sistema.*

Código	Descrição
RF01	O nó sensor deve coletar dados de peso da colmeia (kg).
RF02	O nó sensor deve coletar dados de temperatura interna (°C).
RF03	O nó sensor deve coletar dados de umidade interna (%).
RF04	O nó sensor deve coletar dados de temperatura externa (°C).
RF05	O nó sensor deve obter um <i>timestamp</i> preciso (data/hora) no momento de cada coleta, utilizando um módulo RTC (<i>Real-Time Clock</i>).
RF06	O nó sensor deve formatar os dados coletados (peso, temperaturas, umidade, <i>timestamp</i>) em um <i>payload</i> JSON padronizado.
RF07	O nó sensor deve tentar se conectar a uma rede (<i>Wi-Fi/LoRaWAN</i>) para transmitir o <i>payload</i> JSON a um <i>broker</i> MQTT.
RF08	Caso a conexão com a rede ou com o <i>broker</i> MQTT falhe, o nó sensor deve salvar o <i>payload</i> JSON em um <i>buffer</i> na memória interna.
RF09	Ao restabelecer a conexão, o nó sensor deve enviar prioritariamente todos os dados armazenados no <i>buffer</i> conforme limite de envio antes de enviar a leitura atual.
RF10	O sistema deve permitir o cadastro, autenticação e gerenciamento de usuários.
RF11	O usuário deve poder cadastrar, visualizar, editar e excluir seus apiários.
RF12	O usuário deve poder cadastrar, visualizar, editar e excluir colmeias, associando-as a um apiário.
RF13	A plataforma deve receber os dados (<i>payloads</i> JSON) do <i>broker</i> MQTT e armazená-los no banco de dados, associando-os à colmeia correta.
RF14	O nó sensor deve fornecer uma interface <i>web</i> local (<i>Captive Portal</i>), acessível via <i>Wi-Fi</i> , que permita ao usuário inserir e salvar o SSID, a senha da rede e o ID único da colmeia.

Código	Descrição
RF15	O usuário deve poder visualizar os dados históricos dos sensores (peso, temperatura, umidade) de uma colmeia em gráficos interativos, com filtros por período.
RF16	O usuário deve poder registrar e consultar ações de manejo (ex: troca de rainha, alimentação) associadas a uma colmeia.
RF17	O usuário deve poder registrar e consultar dados de colheitas associadas a um apiário.
RF18	O sistema deve gerar alertas automáticos com base em regras predefinidas (ex: perda de peso abrupta, variações de temperatura).
RF19	O nó sensor deve permitir a execução imediata de uma leitura e envio de dados, interrompendo o estado de repouso mediante um comando físico de curta duração (clique simples no botão).
RF20	O nó sensor deve emitir <i>feedbacks</i> sonoros distintos (via buzzer) para confirmar ao usuário o reconhecimento de ações.
RF21	O usuário poderá exportar o histórico de leituras dos sensores em formato CSV, permitindo análises externas e <i>backup</i> dos dados.
RF22	O usuário poderá visualizar uma lista de alertas gerados, filtrar por status (novo, lido, resolvido) e severidade, além de atualizar o status de cada alerta (marcar como lido ou resolvido).
RF23	O sistema deve apresentar um painel inicial (<i>dashboard</i>) com métricas consolidadas, incluindo total de apiários, status de saúde das colmeias, produção recente e alertas ativos.

Tabela 3.2: Requisitos Não Funcionais do Sistema.

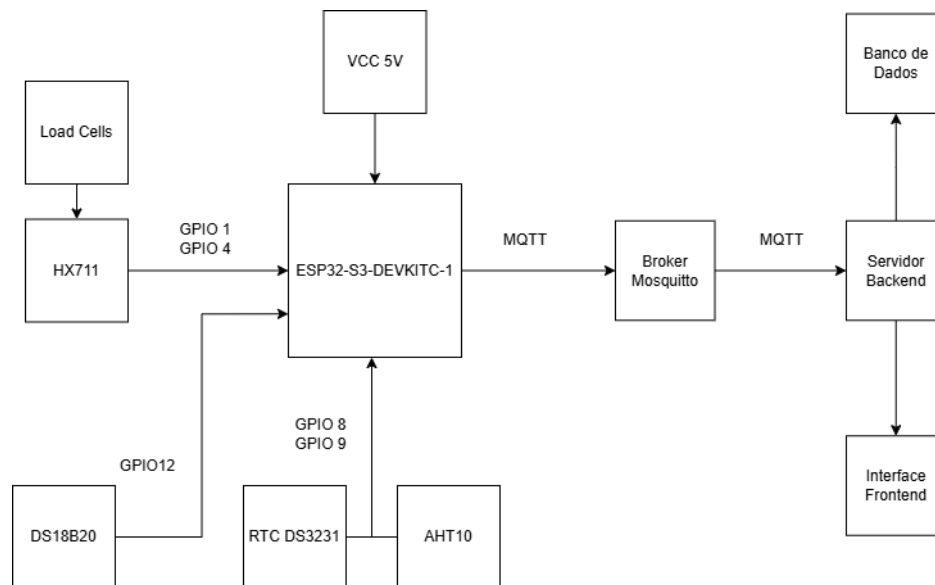
Código	Descrição
RNF01	O nó sensor deve ter consumo de energia otimizado.
RNF02	O sistema deve garantir que nenhuma leitura de sensor seja perdida, mesmo em caso de falha temporária de conexão com a internet.
RNF03	Todos os dados armazenados devem possuir um <i>timestamp</i> preciso para garantir a fidelidade dos eventos.
RNF04	A arquitetura do banco de dados deve ser otimizada para a ingestão e consulta eficiente de séries temporais provenientes de múltiplos dispositivos simultaneamente.
RNF05	O acesso à plataforma <i>web</i> deve ser restrito por autenticação (login e senha), utilizando o padrão <i>JSON Web Token</i> (JWT) para garantir a segurança na comunicação entre o <i>frontend</i> e o <i>backend</i> .
RNF06	A interface do usuário (<i>frontend</i>) deve ser intuitiva e de fácil utilização para apicultores com diferentes níveis de afinidade tecnológica.

Código	Descrição
RNF07	A infraestrutura de <i>backend</i> deve ser containerizada para facilitar a implantação, manutenção e replicação do ambiente de desenvolvimento e produção.
RNF08	O código do <i>backend</i> deve seguir uma arquitetura modular para garantir tipagem estática, facilitando a manutenção e evolução do sistema.

3.2.1 Definição da arquitetura do sistema

A arquitetura do sistema foi desenvolvida seguindo arquiteturas padrão da Internet das Coisas, integrando sensores para aquisição de dados em campo, uma unidade de processamento central e uma infraestrutura de servidor para armazenamento e visualização. A organização da arquitetura pode ser visualizada na Figura 3.1.

Figura 3.1: Diagrama de blocos.



Fonte: Elaborado pelos autores (2025)

3.3 Desenvolvimento do Hardware

O nó sensor representa o componente de *hardware* central do sistema e atua como a unidade responsável pela aquisição de dados em campo. Seu desenvolvimento fundamentou-se nos requisitos de baixo consumo (RNF01), resiliência (RNF02) e na coleta de informações (RF01–RF05).

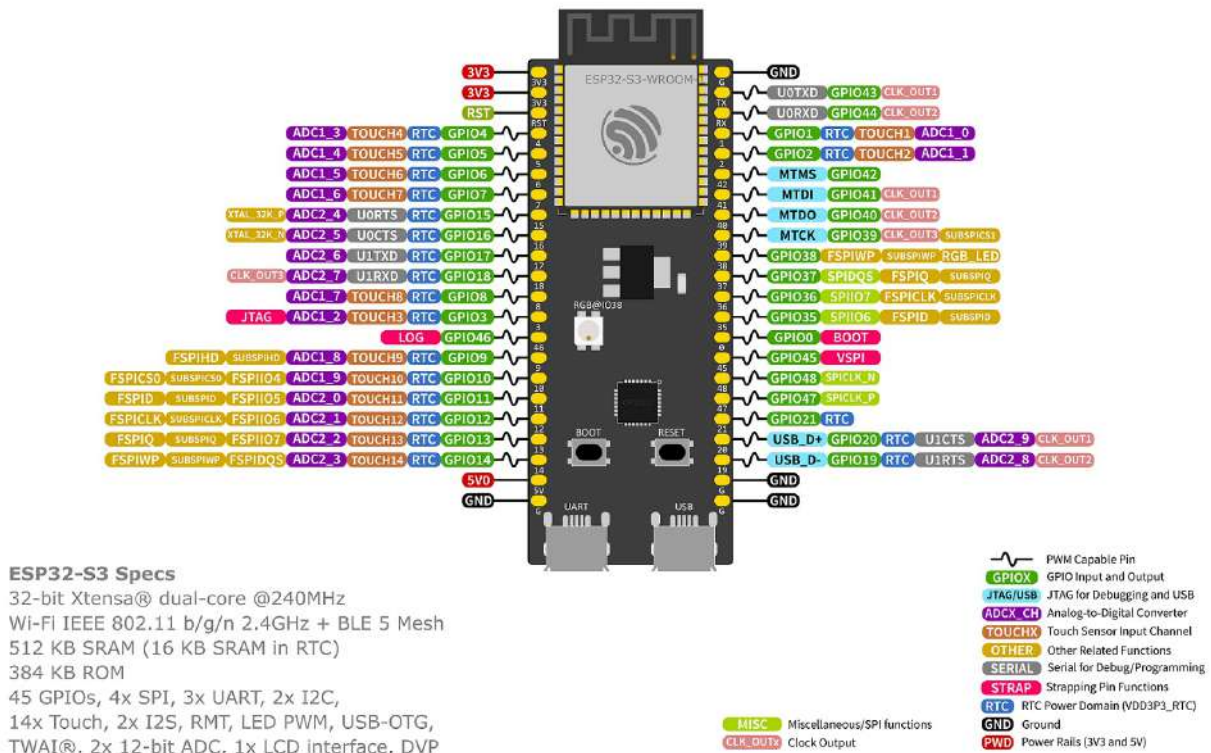
3.3.1 Seleção de Componentes e arquitetura do Hardware

A Placa de Desenvolvimento ESP32-S3 (*DevKit*), tipicamente referida como ESP32-S3-DevKitC-1, é responsável por integrar o microcontrolador ESP32-S3 em uma plataforma pronta

para uso e prototipagem conforme representado na Figura 3.2. Seu principal objetivo é evitar a necessidade de *hardware* de suporte personalizado, permitindo que os desenvolvedores se concentrem exclusivamente no *firmware* e na lógica de aplicação. A *DevKit* facilita a criação rápida de provas de conceito para projetos de IoT. Este componente foi selecionado por sua combinação de processamento dual-core, conectividade *Wi-Fi* nativa (utilizada para a transmissão de dados ao (*gateway/broker*)), ampla quantidade de pinos GPIO e, fundamentalmente, seus modos de *Deep Sleep* de baixo consumo, essenciais para atender ao requisito RNF01, e seu armazenamento 16 MB que atende o RF08.

Figura 3.2: Ilustração e pinout do ESP32-S3 DevKitC-1.

ESP32-S3-DevKitC-1



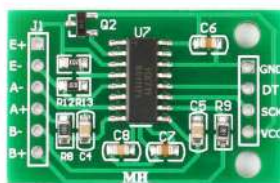
Fonte: EMBARCADOS. (2023)

Para garantir a precisão temporal de cada leitura (RNF03), foi integrado um módulo *Real-Time Clock* (RTC) DS3231, representado pela Figura 3.3. Este componente possui seu próprio oscilador de cristal compensado por temperatura (TCXO) e bateria, permitindo manter a data e hora corretas de forma independente, mesmo quando o ESP32 está em *deep sleep* ou se o dispositivo perde momentaneamente a alimentação principal. A comunicação com o ESP32 ocorre via protocolo I2C.

Figura 3.3: *RTC 3231.*

Fonte: MAKERHERO. (2025)

O HX711 (Figura 3.4) é um conversor A/D de 24 bits com amplificador de instrumentação de baixo ruído integrado, projetado para leitura de sinais diferenciais de *strain gauges* em ponte de *Wheatstone*. Oferece ganhos configuráveis, dois canais de entrada e interface serial de dois fios (DATA e CLK), o que facilita a integração com microcontroladores. A elevada resolução efetiva e a rejeição de ruído de alimentação viabilizam medições precisas de pequenas variações de peso/força em balanças (AVIA SEMICONDUCTOR, 2018).

Figura 3.4: *Módulo HX711.*

Fonte: ORELLANA (2025)

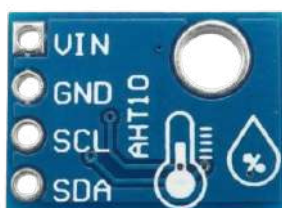
Para o monitoramento de peso (RF01), foi aproveitada uma balança corporal que utiliza quatro células de carga (strain gauge) de 50kg cada, onde é possível visualizar sua forma através da Figura 3.5, dispostas em uma configuração de ponte de *Wheatstone* completa. O sinal analógico diferencial, que é extremamente baixo, é lido, amplificado e convertido pelo módulo HX711. Este módulo é um conversor Analógico-Digital (A/D) de 24 bits de alta precisão, projetado especificamente para esta aplicação, que se comunica digitalmente com o ESP32.

Figura 3.5: *Célula de carga de 50 kg.*

Fonte: Elaborado pelos autores (2025)

O módulo AHT10, representado pela Figura 3.6, é responsável por medir a temperatura e a umidade internas relativas da colmeia (RF02, RF03). Foi escolhido por sua precisão e por utilizar a interface I2C, permitindo a comunicação na mesma linha de barramento do RTC. Ele é amplamente utilizado em projetos com microcontroladores como *Arduino*, *Raspberry Pi* e ESP32. Em teste realizados durante a execução do projeto, ofereceu respostas rápidas com boa estabilidade, com precisão típica de $\pm 2\%$ para umidade relativa e $\pm 0,3^{\circ}\text{C}$ para temperatura, além de operar numa faixa de 0 a 100% de umidade e -40°C a 85°C de temperatura segundo seu fabricante (ARSAIR, 2018).

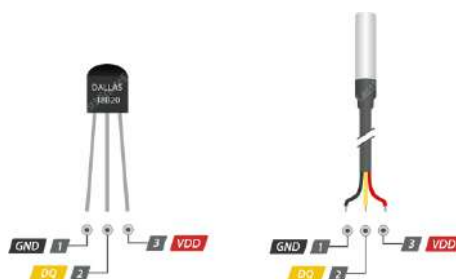
Figura 3.6: Módulo sensor de temperatura AHT10.



Fonte: Elaborado pelos autores (2025)

Responsável por medir a temperatura externa (RF04), o DS18B20 é um sensor digital de temperatura muito utilizado em projetos de eletrônica e IoT. Ele é capaz de medir temperaturas na faixa de -55°C a $+125^{\circ}\text{C}$ com precisão de $\pm 0,5^{\circ}\text{C}$ entre -10°C e $+85^{\circ}\text{C}$, segundo seu fabricante (DALLAS SEMICONDUCTOR, 2018). O sensor funciona com uma interface digital chamada *One-Wire*, que permite a conexão de vários sensores em um único pino de dados de um microcontrolador, facilitando seu uso. Optou-se por este sensor em seu encapsulamento à prova d'água que pode ser visto na Figura 3.7, ideal para a exposição ao ambiente externo, sendo amplamente empregado em aplicações domésticas, industriais e de monitoramento ambiental devido à precisão, facilidade de comunicação e robustez.

Figura 3.7: Módulo sensor de temperatura DS18B20.



Fonte: PRASYO (2018)

O conversor DC-DC MT3608 é um componente essencial para a regulação de energia no sistema, atuando como um conversor do tipo *Step-Up (Boost)*. Sua aplicação é crítica na interface com a célula fotovoltaica, visto que a geração de energia solar apresenta intermitência e flutuações

de tensão decorrentes da variação da incidência luminosa. Frequentemente, a tensão gerada pelo painel é inferior à tensão nominal de operação dos microcontroladores ou do circuito de carga da bateria. O MT3608 soluciona este problema elevando a tensão de entrada para um patamar configurável e estável, garantindo a integridade e a continuidade do fornecimento elétrico aos componentes lógicos do projeto.

A autonomia do sistema de monitoramento depende da eficiência e segurança do armazenamento de energia. Para tal, adotou-se o módulo TP4056 como interface entre a fonte geradora e a bateria. Este dispositivo desempenha dupla função: atua como carregador linear, garantindo que a bateria receba carga dentro dos parâmetros seguros de tensão (máximo de 4,2V) e corrente; e atua como sistema de proteção ativo (*Battery Protection System*). Esta proteção é indispensável em aplicações remotas, pois interrompe o fornecimento de energia caso a tensão da bateria caia abaixo do limiar crítico (geralmente 2,4V), impedindo a degradação química irreversível da célula de lítio e garantindo que o sistema possa ser recarregado automaticamente assim que a fonte de energia (painel solar) estiver ativa novamente. A Tabela 3.3 elenca os componentes utilizados e os valores desembolsados para adquiri-los durante a execução do projeto.

Tabela 3.3: *Relação de componentes do hardware e custo.*

Componente	Custo aproximado
AHT10	R\$ 6,00
Buzzer Passivo	R\$ 1,00
Balança corporal	R\$ 38,00
ESP32 S3	R\$ 42,00
DS18B20	R\$ 12,00
HX711	R\$ 6,00
MT3608	R\$ 3,00
RTC DS3231	R\$ 23,00
TP4056	R\$ 3,00

Fonte: Elaborado pelos autores (2025)

Os componentes foram interligados conforme seus barramentos de comunicação (I2C, *One-Wire* e pinos digitais para o HX711), devidamente representados no diagrama elétrico, representado pela Figura 3.8.

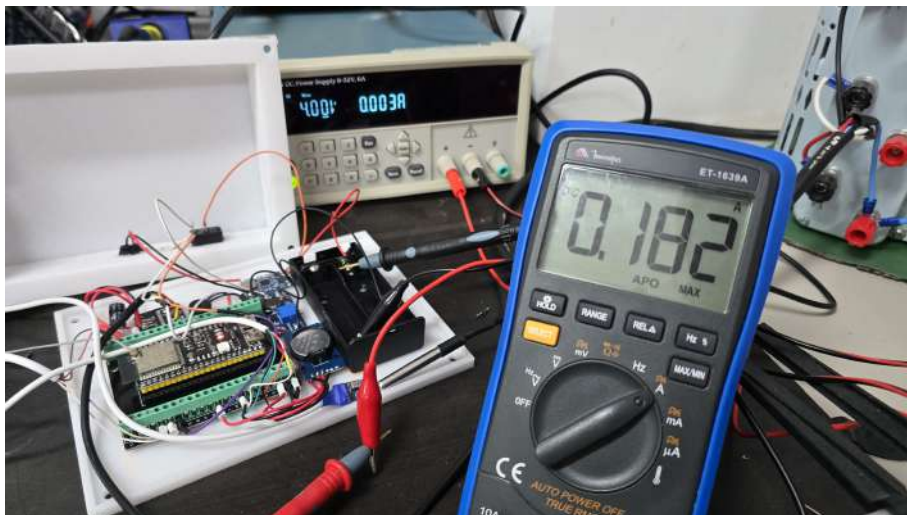
temperatura e estabilidade de superfície, onde foi necessário compreender conceitos relacionados sobre extensômetros, em que tais condições são cruciais para o funcionamento correto das células de carga. Com o *hardware* montado e o *firmware* básico implementado, a função `load_cell.tare()` foi executada para zerar a balança. Em seguida, pesos de massa conhecida (1kg, 5kg, 20kg) foram posicionados sobre a balança, além da adição progressiva no volume de água em um recipiente, visto que a densidade da água tem o valor igual a 1. O valor `CALIBRATION_FACTOR` (definido no firmware) foi ajustado experimentalmente até que a leitura fornecida pela função `load_cell.get_units()` correspondesse ao mais próximo do peso real aplicado, garantindo a acurácia das medições.

3.3.2 Consumo elétrico

O protótipo conta com 02 baterias de lítio do modelo 18650 conectadas em paralelo, com capacidade nominal de 2850 mAh, totalizando 5700 mAh, que podem ser recarregadas através da célula fotovoltaica que integra o projeto.

O consumo máximo do protótipo alcança 200 mA a cada coleta ao considerar uma margem de erro de aproximadamente 10%, conforme exposto na Figura 3.9, com a utilização do modo de *deep sleep* em que o protótipo consome apenas 3 mA, em que as coletas foram programadas para ocorrer a cada 1 hora, ou seja, a capacidade das baterias associada ao modo *deep sleep*, considerando que o dispositivo acorde por 10 segundos, o consumo por hora é de aproximadamente 3,56 mA.

Figura 3.9: *Teste de consumo máximo e mínimo.*



Fonte: Elaborado pelos autores (2025)

Em caso de falha no carregamento das baterias e considerando o consumo mensurado na Figura 3.9 teremos uma autonomia teórica de:

$$Autonomia = \frac{5700}{3,56} = 1603h$$

Portanto, nas condições mencionadas, o dispositivo tem uma autonomia teórica de 1603 horas, que resulta também em 1603 coletas, por ocorrer 1 coleta por hora, totalizando aproximadamente 67 dias sem carregamento.

3.3.3 Implementação do módulo LoRa

Durante a fase de desenvolvimento do protótipo, optou-se pela realização de testes práticos com a tecnologia de comunicação LoRa, visando validar sua capacidade teórica de transmissão de dados a longas distâncias com baixo consumo energético. No entanto, o ensaio revelou incompatibilidades críticas tanto no âmbito regulatório quanto no desempenho técnico do *hardware* adquirido. O módulo LoRa utilizado nos testes operava em uma faixa de frequência não licenciada para uso em território nacional, visto que foi comprado em uma plataforma de *e-commerce* estrangeira. O dispositivo adquirido é configurado para operar na frequência 433 MHz, destinada a outras regiões geográficas como, partes da Ásia e da Europa, divergindo do padrão estabelecido pela Agência Nacional de Telecomunicações (ANATEL). No Brasil, a operação de dispositivos LoRa deve ocorrer, majoritariamente, na faixa de frequência de 915 MHz a 928 MHz (plano de frequências AU915) (ALFA COMP, s.d.). A utilização de equipamentos fora da especificação da ANATEL pode causar problemas graves com a operação em faixas não permitidas, como interferência prejudicial em serviços primários de telecomunicação e outros sistemas críticos licenciados no país. No que tange ao desempenho, a literatura técnica sugere que módulos LoRa, em condições de visada direta, podem atingir alcances na ordem de quilômetros. Entretanto, nos testes realizados, observou-se uma atenuação drástica do sinal. O dispositivo, que prometia cobertura ampla, apresentou perda de conexão estável a distâncias superiores a 30 metros. Diante da inviabilidade técnica comprovada pela baixa eficiência de transmissão e a ausência de tempo hábil para adquirir o equipamento correto, optou-se pela descontinuação deste módulo específico no projeto, redirecionando o desenvolvimento para soluções mais acessíveis naquele momento, como a utilização de redes *Wi-Fi* apenas, com o auxílio do protocolo MQTT.

3.4 Desenvolvimento do *Firmware*

O *firmware* do nó sensor utiliza as linguagens C/C++ no ecossistema *Arduino*, com execução no microcontrolador ESP32-S3. A adoção desta plataforma e de seu ambiente de desenvolvimento (*IDE*) justifica-se pela ampla disponibilidade de bibliotecas, que simplificam a comunicação com periféricos e abstraem camadas de baixo nível do *hardware*. A integração desses componentes permite a implementação de uma arquitetura de eventos e estados, projetada especificamente para o atendimento de requisitos fundamentais: economia de energia (RNF01), resiliência diante de falhas de conexão (RNF02) e facilidade de configuração em campo.

3.4.1 Lógica de Execução e Gerenciamento de Energia (*Deep Sleep*)

Para atender ao requisito de baixo consumo energético (RNF01), o desenvolvimento do *firmware* adota uma abordagem distinta do paradigma convencional do ecossistema *Arduino*. Em vez da execução contínua em um laço de repetição infinito (`loop()`), toda a lógica de operação é processada de maneira linear e finita dentro da função de configuração (`setup()`), encerrando-se na chamada da função de repouso profundo (`go_to_sleep()`). Esse gerenciamento é reforçado pela estratégia de *Power Gating*, na qual o *firmware* controla transistores via pinos digitais para interromper fisicamente a alimentação de sensores e periféricos ociosos, garantindo consumo residual nulo desses componentes durante o intervalo de espera.

O ciclo de vida do dispositivo é disparado pela identificação da causa do despertar, que define o comportamento inicial do sistema: um despertar por temporizador, ocorrendo em intervalos programados para coleta automática, ou um despertar externo, acionado por interação física do usuário para rotinas de manutenção. Independentemente da origem, a função de configuração orquestra uma sequência rigorosa de operações que inicia com a configuração de periféricos e do sistema de arquivos, seguida pela recuperação de estados e calibrações armazenados em memória não volátil.

Definido o modo de operação, seja pelo intervalo agendado no temporizador ou pelo acionamento do botão físico, o sistema realiza a energização dos sensores para a aquisição imediata dos dados de peso e clima. Somente após a conclusão da coleta, é iniciada a tentativa de conexão à rede *Wi-Fi* e ao *broker* MQTT, seguida da sincronização de horário via NTP para garantir a precisão do *timestamp*. O processo encerra-se com o envio das leituras, ou seu armazenamento em *buffer* interno em caso de falha de conexão, e o desligamento controlado de todos os periféricos antes da reentrada em modo *Deep Sleep*.

3.4.2 Modos de Operação e Interface

Para atender à necessidade de operação simplificada em campo, foi implementada uma lógica de botão multifuncional baseada no tempo de pressionamento, com *feedback* sonoro via *buzzer* passivo para orientar o usuário sem a necessidade de *displays* visuais:

1. **Leitura Forçada (Clique Curto):** Acorda o dispositivo fora do horário agendado para realizar uma medição imediata e envio de dados. Útil para verificar o funcionamento após uma instalação.
2. **Tara da Balança (3 a 6 segundos):** O sistema zera a balança (tara), salvando o novo *offset* na memória permanente (*Preferences*). O *feedback* sonoro é um bip duplo repetitivo.
3. **Modo de Configuração (Mais de 6 segundos):** O dispositivo cria uma rede *Wi-Fi* própria ("Colmeia-Config"). Ao se conectar a ela pelo celular, o apicultor acessa uma interface *web* onde pode configurar o nome da rede *Wi-Fi* local e o ID da colmeia.

3.4.3 Coleta, Formatação e *Timestamping*

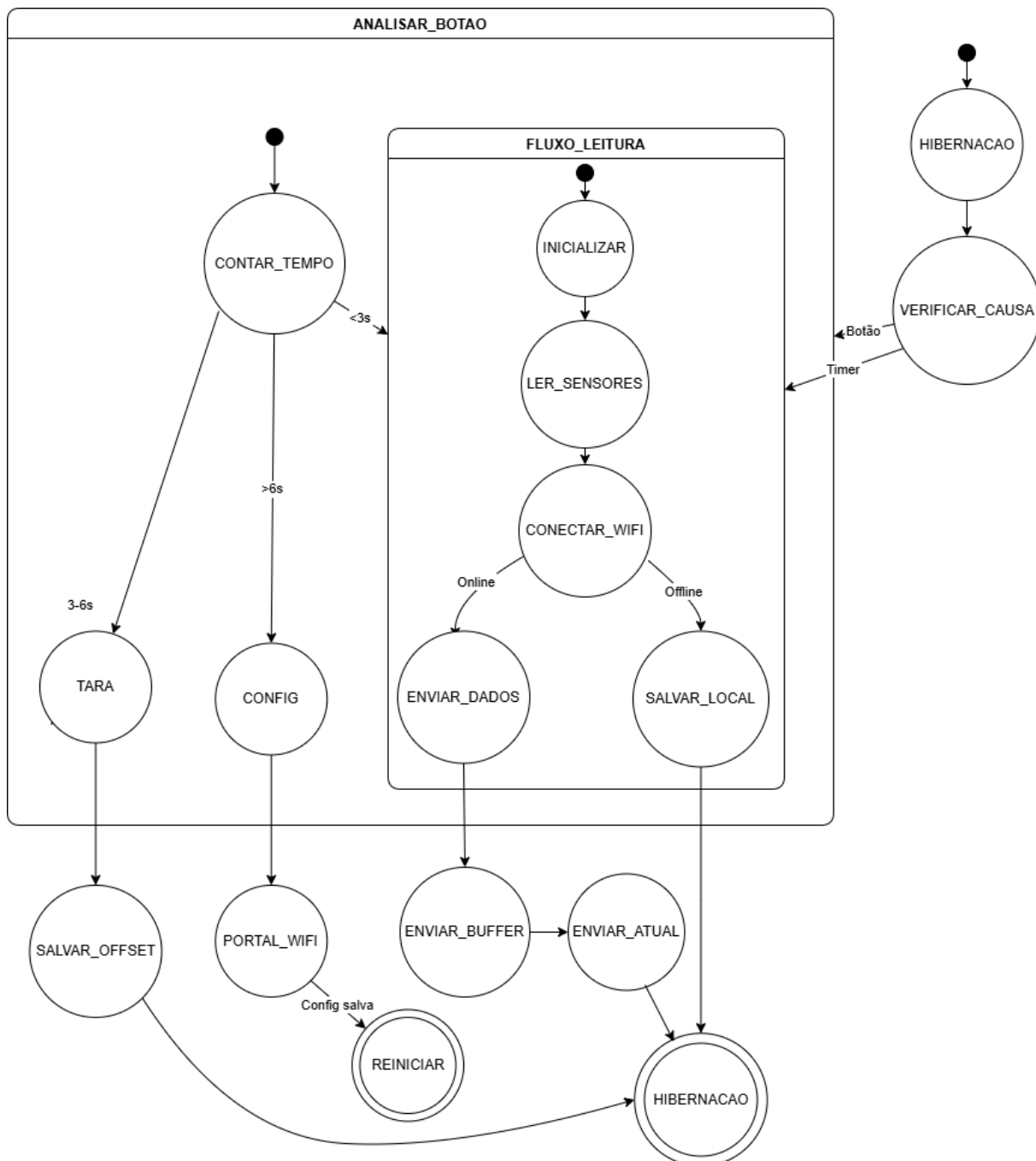
A integridade temporal dos dados é assegurada por um sistema híbrido de cronometragem. O dispositivo utiliza um módulo de relógio de tempo real (*Real Time Clock* - RTC DS3231) para registrar o momento exato de cada leitura. Em situações de falha ou desatualização do módulo, o sistema realiza a sincronização via protocolo NTP (*Network Time Protocol*), desde que haja conectividade com a internet, atualizando o *hardware* RTC para garantir a precisão de registros futuros.

Os dados são estruturados em um *payload* no formato *JavaScript Object Notation* (JSON) , garantindo a padronização para o consumo pela plataforma de nuvem. A estrutura da mensagem enviada segue o modelo exemplificado abaixo:

```
{  
  "weight": "25.350",  
  "temp_i": "34.80",  
  "humid_i": "65.50",  
  "temp_e": "28.12",  
  "timestamp": "2025-10-29T15:30:00Z"  
}
```

3.4.4 Conectividade e Resiliência

A integridade das informações em cenários de instabilidade ou ausência de rede é assegurada por um mecanismo de resiliência baseado em armazenamento temporário (*buffer*). O funcionamento desse sistema é coordenado por uma máquina de estados, ilustrada na Figura 3.10, que define o fluxo de transmissão conforme a disponibilidade de conexão com o *broker* MQTT.

Figura 3.10: Representação do sistema em máquina de estados.

Fonte: Elaborado pelos autores (2025)

Ao estabelecer a comunicação, o dispositivo verifica a existência de registros pendentes e realiza o envio dessas informações em lotes. Como o sistema de arquivos interno não realiza o armazenamento de forma necessariamente sequencial, a consistência cronológica da série histórica é garantida pelo *timestamp* individual de cada leitura, permitindo a ordenação correta dos dados no momento da integração com o banco de dados.

Caso a conexão não seja estabelecida, a leitura atual é estruturada em formato JSON e armazenada internamente, seguida pela entrada imediata em modo *Deep Sleep*. Essa abordagem assegura

que, mesmo com a aleatoriedade na ordem de envio dos arquivos do *buffer*, a plataforma final consiga reconstruir o histórico da colmeia sem perda de continuidade.

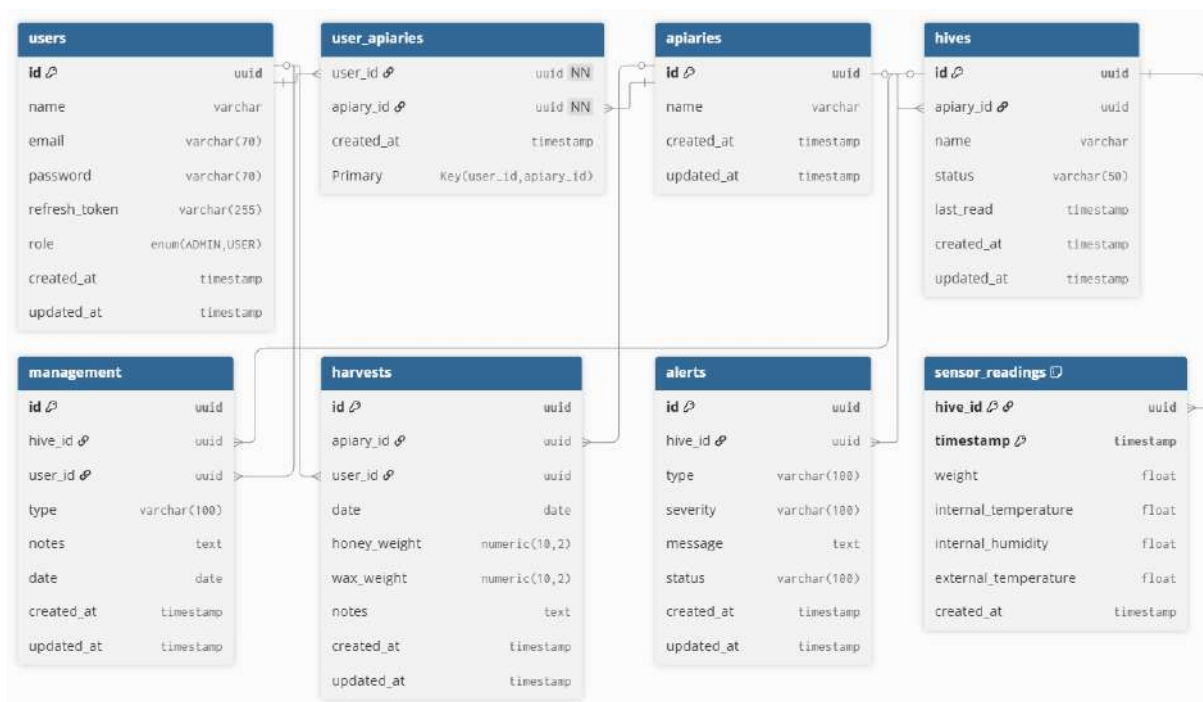
3.5 Banco de Dados e Mensageria

O banco de dados compõe a camada de persistência da plataforma de nuvem, atuando como o repositório para todos os dados gerados tanto pelos nós sensores quanto pelos usuários. A implementação desta base de dados foi orientada pelo requisito de escalabilidade (RNF04), visando uma estrutura capaz de lidar com a ingestão constante de dados de séries temporais. Para atender à complexidade do sistema, que demanda o gerenciamento de dados relacionais clássicos, como perfis de usuários e registros de colmeias, simultaneamente às leituras frequentes de sensores, adotou-se uma solução híbrida baseada no *PostgreSQL*.

Este Sistema de Gerenciamento de Banco de Dados (SGBD) foi selecionado devido à sua robustez, confiabilidade e conformidade com o padrão SQL. Integrada a ele, utiliza-se a *TimescaleDB*, uma extensão de código aberto que otimiza o *PostgreSQL* especificamente para cargas de trabalho de séries temporais. Essa escolha técnica visa garantir a eficiência no armazenamento e na consulta da série histórica de leituras, assegurando a performance do sistema à medida que o volume de registros aumenta.

3.5.1 Modelagem e Estrutura dos Dados

A estrutura do banco de dados foi modelada para refletir as entidades do mundo real e seus relacionamentos, garantindo a integridade referencial. As tabelas estão organizadas conforme o Diagrama Entidade-Relacionamento apresentado na Figura 3.11.

Figura 3.11: Modelagem relacional

Fonte: Elaborado pelos autores (2025)

3.5.2 Otimização de Séries Temporais (*TimescaleDB*)

A tabela `sensor_readings` foi projetada para cumprir o requisito RF13, armazenando as leituras individuais de peso, temperatura interna, temperatura externa e umidade. Devido à natureza contínua da coleta de dados e ao potencial de crescimento da rede de monitoramento, optou-se por uma estrutura que favorece a eficiência em longo prazo. Utilizando a extensão *TimescaleDB*, a tabela foi convertida em uma *Hypertable*, permitindo que o sistema gerencie o acúmulo de dados de forma otimizada.

Nessa arquitetura, o *TimescaleDB* realiza o particionamento automático dos registros em segmentos temporais, denominados *chunks*, com base na coluna *timestamp*. Essa organização oferece vantagens fundamentais para a operação do sistema, como a manutenção de uma alta performance de ingestão, visto que o banco de dados prioriza a escrita no segmento de tempo mais recente. Além disso, as consultas destinadas aos *dashboards*, como a análise histórica do peso de uma colmeia específica, tornam-se consideravelmente mais rápidas, pois o motor de busca processa apenas os segmentos de tempo relevantes em vez de percorrer a totalidade dos registros.

Essa abordagem na estruturação dos dados assegura a escalabilidade (RNF04) da plataforma. Mesmo que o sistema inicie com um número reduzido de dispositivos, a arquitetura está preparada para suportar o adensamento dos dados provenientes de ciclos de coleta frequentes e da expansão para novos apiários.

3.5.3 Mensageria com Eclipse Mosquitto

Para a composição da camada de mensageria, utiliza-se o *broker Eclipse Mosquitto* com o intuito de intermediar a comunicação entre os nós sensores e a *Application Programming Interface* (API) de gerenciamento. A escolha deste componente fundamenta-se em sua leveza e alta eficiência no processamento de mensagens baseadas no protocolo MQTT, características consideradas essenciais para sistemas de monitoramento *IoT* que operam em ambientes sujeitos a instabilidades de conexão. O *Mosquitto* atua como o ponto central de distribuição de dados, operando de forma desacoplada para garantir que as informações coletadas no campo sejam recebidas e enfileiradas corretamente. Essa estrutura assíncrona assegura que o fluxo de dados seja mantido de forma íntegra antes de seu processamento e persistência final na base de dados, permitindo que o sistema suporte múltiplas conexões simultâneas.

3.6 Desenvolvimento *backend* (API)

O *backend* representa a camada de aplicação da plataforma, sendo responsável por centralizar as operações lógicas e a integração entre os componentes do sistema. Esta camada desempenha funções essenciais para o funcionamento da solução, como a ingestão e o processamento de dados provenientes dos dispositivos *IoT* via protocolo MQTT, além da disponibilização de uma interface de programação no padrão *REST* para a comunicação com o *frontend*.

Adicionalmente, o *backend* gerencia as regras de negócio e as interações com o sistema de banco de dados, garantindo a persistência e a integridade das informações coletadas. O desenvolvimento deste módulo foi norteado pelos requisitos de segurança (RNF05), escalabilidade (RNF04) e portabilidade (RNF07), assegurando que a infraestrutura suporte o crescimento da rede de sensores e mantenha a integração entre os serviços da plataforma.

3.6.1 Seleção de Tecnologias

Para a construção do *backend*, adotou-se um conjunto de tecnologias focado em escalabilidade, tipagem forte e agilidade no desenvolvimento. O ambiente de execução *Node.js* foi selecionado por sua natureza assíncrona e orientada a eventos, o que o torna ideal para lidar com operações de entrada e saída (I/O) de alta concorrência, como as conexões MQTT e as múltiplas requisições da API. Sobre essa base, utilizou-se o *framework Nest.js* em conjunto com a linguagem *TypeScript*, o que permitiu o estabelecimento de uma arquitetura modular e a redução de erros por meio da tipagem estática. No que se refere à integração com a camada de persistência, empregou-se o *TypeORM* como ferramenta de mapeamento objeto-relacional, simplificando a comunicação entre a lógica da aplicação e o banco de dados *PostgreSQL*.

A transparência e a padronização da comunicação com o *frontend* foram garantidas pela especificação *OpenAPI* com a ferramenta *Swagger*, que gera automaticamente a documentação interativa dos *endpoints*. Além das ferramentas de codificação, o fluxo de trabalho foi estruturado com o

uso do *Git* e da plataforma *GitHub* para o versionamento e controle colaborativo do código-fonte. Por fim, a utilização do *Docker* foi fundamental para assegurar a portabilidade da aplicação, permitindo que o ambiente de execução fosse replicado de forma idêntica desde o desenvolvimento local até a implantação final no servidor de produção.

3.6.2 Arquitetura Modular

A estruturação do *backend* fundamenta-se nos princípios de arquitetura modular do *framework* *Nest.js*, em que a aplicação é organizada em unidades funcionais com responsabilidades únicas. Além dos módulos básicos para operações de cadastro e manutenção de usuários, apiários, colmeias e colheitas, o sistema integra módulos especializados para agregação de inteligência. O *AuthModule* centraliza a segurança e o controle de acesso por meio de JWT, enquanto o *SensorReadingsModule* operacionaliza a persistência de dados e a exportação de históricos em formato CSV, conforme o requisito RF21.

Complementarmente, a lógica de monitoramento é composta pelo *DashboardModule*, que consolida indicadores estatísticos para o painel do usuário, e pelo *AlertModule*, responsável pela detecção de anomalias nos dados coletados. A manutenção automatizada do sistema é coordenada pelo *TaskModule* através de tarefas agendadas, enquanto o *MqttModule* assegura a ingestão contínua de dados provenientes do *broker*. Essa organização facilita a escalabilidade e a manutenção do sistema (RNF08), permitindo que novas funcionalidades sejam acopladas de forma independente, preservando a integridade das camadas existentes.

3.6.3 Comunicação e Ingestão de Dados (MQTT)

O *MqttModule* atua como o ponto de entrada para as informações provenientes do *hardware*, atendendo ao requisito RF13. Este módulo estabelece uma conexão persistente com o servidor *Mosquitto* e realiza a subscrição no tópico de telemetria para a recepção das leituras. Ao processar o *payload* JSON enviado pelo dispositivo ESP32, o serviço utiliza o *SensorReadingsService* para validar as informações e encaminhá-las para a persistência na hipertabela do *TimescaleDB*. Esse fluxo garante que o dado bruto seja tratado e armazenado com precisão, servindo de base para as análises temporais da plataforma.

3.6.4 Interface de API, Segurança e Documentação

À exceção do *MqttModule*, todas as funcionalidades de negócio são expostas por meio de uma interface de programação de aplicações no padrão *REST*. Para atender aos requisitos de segurança e controle de acesso (RNF05), o sistema integra o *AuthModule*, que implementa uma estrutura de autenticação baseada em JWT. Esse fluxo garante que as requisições subsequentes ao *login* sejam validadas por mecanismos de proteção, conhecidos como *Guards*, os quais asseguram que os usuários acessem exclusivamente os dados pertinentes aos seus perfis, impedindo interações não autorizadas com registros de terceiros.

A integridade das informações trafegadas nesta interface é garantida pela implementação de Objetos de Transferência de Dados, os DTOs (*Data Transfer Objects (DTO)*), que estabelecem uma estrutura rigorosa para a entrada e saída de dados. A utilização da biblioteca *class-validator* complementa esse processo, permitindo a validação automática das informações e assegurando a conformidade técnica antes do processamento pelas camadas de negócio.

Para viabilizar a manutenção e a integração entre os serviços, utiliza-se a especificação *OpenAPI* por meio da ferramenta *Swagger*. Essa solução gera uma documentação interativa que detalha os *endpoints*, parâmetros de consulta e os modelos de resposta esperados, o que promove a transparência e a agilidade no desenvolvimento do *frontend*. Tal padronização é fundamental para assegurar a consistência das comunicações e a eficiência na integração de todos os serviços da plataforma de nuvem.

3.6.5 Automação e Detecção de Anomalias

Para elevar o nível de inteligência da solução, implementaram-se mecanismos de processamento em segundo plano que operam de forma autônoma, sem a necessidade de intervenção humana direta. No que se refere ao monitoramento de status, utiliza-se um processo agendado (*Cron Job*) executado periodicamente para verificar a latência de comunicação das colmeias. Caso se identifique que um dispositivo não enviou dados por um período superior a 24 horas, o sistema atualiza automaticamente o estado da colmeia para *offline* e gera um alerta crítico de conectividade, informando ao usuário sobre a possível interrupção na coleta.

Paralelamente, a detecção de anomalias é realizada de forma contínua a cada nova leitura recebida pelo sistema. Por meio de um módulo dedicado, os dados coletados são analisados em busca de padrões irregulares, garantindo-se o controle para evitar a duplicação de notificações redundantes. Essas verificações abrangem a identificação de falhas físicas nos sensores, o monitoramento de faixas de segurança para temperatura e umidade, bem como a observação de variações bruscas de peso. Esses parâmetros são fundamentais para a gestão do apiário, pois permitem identificar desde falhas técnicas até eventos biológicos ou externos, como processos de enxameação ou ocorrência de pilhagens, possibilitando uma resposta ágil por parte do apicultor.

3.6.6 Infraestrutura, *DevOps* e Implantação

A infraestrutura do projeto foi projetada para suportar um ciclo de vida de desenvolvimento de *software* moderno, priorizando a automação, a reprodutibilidade de ambientes e a entrega contínua. Todo o código-fonte do *backend* é versionado com a ferramenta Git e hospedado em um repositório no *GitHub*, utilizando a estratégia de *branching* baseada no *Git Flow* para organizar o desenvolvimento colaborativo.

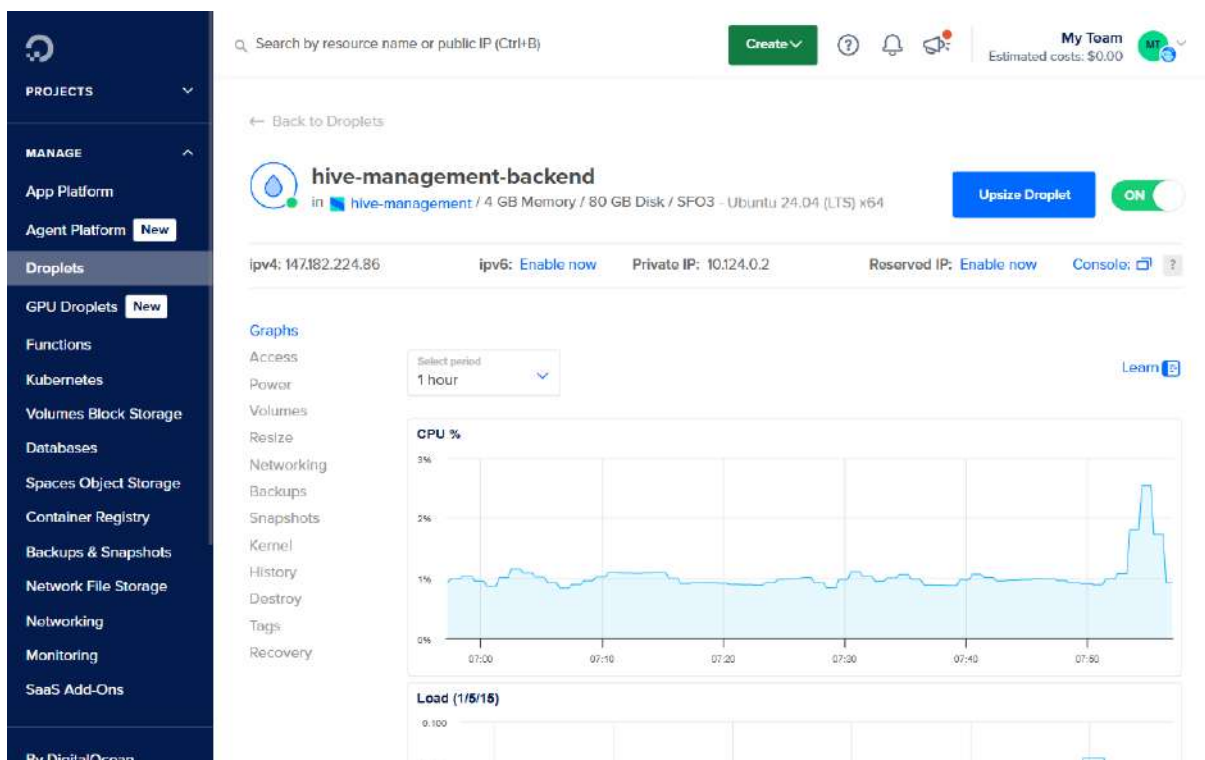
Para automatizar o processo de *build* e *deploy*, implementou-se um *pipeline* de CI/CD utilizando *GitHub Actions*. Este fluxo de automação, acionado a cada *push* na *branch* principal, executa etapas sequenciais que compreendem o *checkout* do código, a instalação de dependências via NPM

e a construção de uma imagem *Docker* otimizada. O processo encerra-se com o *deploy* automático, no qual o *pipeline* estabelece uma conexão SSH com o servidor de produção, realiza a atualização da imagem e reinicia os serviços sem a necessidade de intervenção manual.

Atendendo ao requisito de portabilidade (RNF07), a aplicação utiliza a containerização com *multi-stage build*, segregando as dependências de desenvolvimento das de produção. O orquestrador *Docker Compose* gerencia a execução simultânea do ecossistema, instanciando a API NestJS, o banco de dados *PostgreSQL* e o *broker Eclipse Mosquitto*. Adicionalmente, o *Nginx* atua como *Reverse Proxy*, gerenciando o tráfego de entrada, a terminação SSL (HTTPS) e o roteamento de requisições.

O ambiente de produção está hospedado em uma instância virtual (*Droplet*) na plataforma *DigitalOcean*, como ilustrado na Figura 3.12. A opção por uma VPS (*Virtual Private Server*) permitiu um controle granular sobre a configuração do sistema operacional Linux Ubuntu, do *firewall* e dos recursos de rede, fatores essenciais para garantir a estabilidade da conexão MQTT persistente e a segurança da plataforma.

Figura 3.12: Máquina Virtual (*Droplet*) na *Digital Ocean*



Fonte: Elaborado pelos autores (2025)

3.7 Desenvolvimento da Interface do Usuário (*frontend*)

A interface do usuário (*frontend*) é o componente do sistema com o qual o apicultor interage diretamente. Seu desenvolvimento foi centrado em traduzir os dados provenientes do *backend*, em uma experiência visual clara, interativa e de fácil utilização, cumprindo o requisito de usabilidade

(RNF06).

3.7.1 Seleção de Tecnologias

Para a construção de uma interface moderna, performática e de fácil manutenção, selecionou-se um conjunto de tecnologias baseado na biblioteca *React.js* e na linguagem *TypeScript*. A interface foi desenvolvida sob a arquitetura de uma aplicação de página única integrada à ferramenta de construção *Vite*, escolhida por oferecer tempos de compilação reduzidos e otimizar a produtividade durante o desenvolvimento. O roteamento no lado do cliente é gerenciado pela biblioteca *React Router*, proporcionando uma navegação fluida entre os módulos do sistema. O uso do *TypeScript* garantiu a consistência com o *backend*, assegurando a segurança de tipos e facilitando a integração com os Objetos de Transferência de Dados (*DTOs*). Para a comunicação via protocolo HTTP, adotou-se a biblioteca *Axios*, que permite a integração entre a interface e os serviços de *backend*.

Para a estilização, utilizou-se a combinação do *TailwindCSS* com a biblioteca de componentes *shadcn/ui*, o que acelerou a criação de uma interface acessível e responsiva, abrangendo elementos como modais, tabelas e notificações temporárias. Para cumprir o requisito funcional de visualização histórica (RF15), utilizou-se a biblioteca *Recharts*, responsável pela renderização dos gráficos interativos que correlacionam as variáveis de peso, temperatura e umidade ao longo do tempo. Por fim, a manipulação de registros temporais foi operacionalizada por meio da ferramenta *Day.js*, garantindo que a formatação de todas as datas e o fuso horário estejam em conformidade com o padrão brasileiro (PT-BR).

3.7.2 Arquitetura de Comunicação e Gerenciamento de Estado

A comunicação com o *backend* e o gerenciamento do estado da aplicação constituem os pilares da arquitetura do *frontend*. Para o consumo da API no padrão *REST*, utilizou-se a biblioteca *Axios*, na qual configurou-se um interceptador de requisição. Esse mecanismo anexa automaticamente o *token JWT* ao cabeçalho de autorização em todas as chamadas destinadas a rotas protegidas, garantindo a comunicação segura sem a necessidade de intervenção manual em cada componente. O controle de acesso é reforçado por um fluxo de renovação de credenciais, implementado por meio de um interceptador de resposta para o tratamento de erros de autenticação. Ao detectar a expiração de um *token*, o sistema executa de forma transparente a solicitação de uma nova chave ao *backend* e reenvia a requisição original. Caso o processo de renovação falhe, a sessão do usuário é encerrada e ocorre o redirecionamento automático para a tela de acesso, assegurando o cumprimento do requisito de segurança RNF05.

No que se refere ao gerenciamento de estado global, selecionou-se a biblioteca *Zustand* devido à sua simplicidade e reduzida complexidade de código. Essa ferramenta armazena e disponibiliza os dados de autenticação e as informações do usuário para todos os componentes da plataforma de forma centralizada. Complementarmente, a integridade dos dados em formulários decorre da integração entre as bibliotecas *react-hook-form* e *Zod*. Essa combinação permite o gerenciamento

eficiente dos campos e a definição de esquemas de validação rigorosos, o que garante a conformidade das informações submetidas à API com o formato técnico esperado.

3.7.3 Funcionalidades e Telas Principais

A arquitetura de navegação foi projetada para refletir o fluxo de trabalho natural de um apicultor, abrangendo desde a autenticação inicial até a análise de dados telemétricos. No que tange ao acesso e identificação, o sistema dispõe de interfaces dedicadas para o cadastro de novos usuários, onde se utiliza a biblioteca *zod* para a validação rigorosa de credenciais e formatos de *e-mail*. Realiza-se, junto ao *backend*, uma verificação de integridade para impedir a duplicidade de registros, enquanto o acesso seguro é garantido por meio de autenticação baseada em *tokens* JWT gerenciados em memória. Complementarmente, a rota de perfil permite a confirmação da identidade e dos dados cadastrais do usuário autenticado.

Ao ingressar na plataforma, o usuário é direcionado ao *dashboard*, interface concebida para oferecer consciência situacional imediata sobre a saúde da operação. Esta tela centraliza o monitoramento de alertas por severidade (RF18) e apresenta um diagnóstico de conectividade das colmeias, classificando-as conforme o estado de saúde e atividade. Adicionalmente, são exibidos resumos de produtividade de mel e cera referentes aos últimos 30 dias, além de métricas administrativas que consolidam o total de apiários e a frequência de manejos realizados nos últimos 15 dias. Para o tratamento de anomalias, implementou-se um sistema de notificações rápidas e uma máquina de estados de incidentes, que permite gerenciar o ciclo de vida dos alertas desde a triagem inicial até o histórico de resoluções, servindo como log de auditoria para análises posteriores.

A organização lógica dos ativos fundamenta-se na gestão de apiários e colmeias. Por meio de painéis de controle estruturados em tabelas, viabiliza-se o gerenciamento completo de apiários (RF11) e das respectivas colmeias (RF12), permitindo a realização de cadastros, edições e exclusões de registros de forma centralizada. Destaca-se, nesta etapa, a inclusão de recursos de usabilidade para facilitar a configuração do *hardware*: o sistema disponibiliza um recurso de cópia do ID único da colmeia, garantindo a associação precisa entre o dispositivo *IoT* físico e o respectivo registro no banco de dados.

A visualização técnica das informações ocorre ao acessar os detalhes de uma colmeia específica (RF15), ambiente no qual os dados coletados no campo são apresentados de forma intuitiva. Indicadores em tempo real exibem as últimas leituras de peso, temperatura e umidade, enquanto gráficos interativos permitem acompanhar a evolução das variáveis ao longo do tempo. Com o intuito de oferecer flexibilidade na investigação de períodos determinados, integraram-se ferramentas de filtragem temporal e a funcionalidade de exportação de dados em formato CSV, o que facilita o manuseio das informações em planilhas externas. Adicionalmente, o sistema promove a digitalização do registro operacional por meio das interfaces de registros de colheitas e manejos (RF16). Tais recursos permitem documentar intervenções humanas, como alimentações ou trocas de rainha, consolidando uma base de dados que auxilia na gestão e no controle administrativo da produção.

3.7.4 Experiência do Usuário e Paginação no Lado do Servidor

Para garantir uma experiência de usuário satisfatória (RNF06) e alto desempenho (RNF04), mesmo diante de grandes volumes de dados, o sistema adota estratégias específicas de navegação e visualização. A paginação no lado do servidor (*server-side*) é aplicada às tabelas principais, como apiários, colmeias, colheitas e manejos. Nesse modelo, o *frontend* encaminha os parâmetros de página, limite e ordenação por meio de parâmetros de busca na URL, de modo que a interface renderiza apenas os dados e as meta-informações retornados pela API. Este mecanismo garante que os critérios de visualização permaneçam sincronizados com o navegador, o que possibilita ao usuário compartilhar *links* ou atualizar a página sem perder o contexto dos dados filtrados.

A resposta visual da interface foi planejada para mitigar a percepção de latência durante o carregamento de informações. O sistema utiliza componentes de esqueleto (*skeletons*) enquanto os dados das tabelas e gráficos são recuperados do servidor. Além disso, notificações temporárias informam sobre o sucesso ou erro das operações, enquanto ações de exclusão exigem confirmação em janelas de diálogo, medida que previne erros acidentais e reforça a usabilidade da plataforma.

3.7.5 Infraestrutura, Hospedagem e Entrega Contínua

Diferente do *backend*, que demanda um servidor persistente para as conexões MQTT e o banco de dados, o *frontend* foi projetado como conteúdo estático de alta disponibilidade. Para a hospedagem da interface, selecionou-se a plataforma *Vercel*, que distribui a aplicação por meio de uma *Content Delivery Network* (CDN) global. Esse modelo garante que os arquivos HTML, CSS e *Javascript* sejam entregues a partir de servidores fisicamente próximos ao usuário final, o que atende ao requisito de desempenho RNF04.

O processo de integração e entrega contínua (CI/CD) ocorre de forma automatizada e integrada ao *GitHub*. A cada envio de código para a ramificação principal, um gatilho aciona o fluxo de construção no ambiente da *Vercel*, responsável por transpilar o código *TypeScript* e *React* para arquivos otimizados. Durante essa etapa, o sistema realiza a injeção da variável de ambiente que configura o endereço da API de produção. Após a finalização da construção, a implantação da nova versão é imediata, sem interrupções no serviço.

A estratégia de domínios e o gerenciamento de rede reforçam a segurança do ecossistema. O domínio principal foi registrado na *Hostinger*, mas a autoridade de DNS e a segurança de borda foram delegadas à *Cloudflare*. Esta camada intermediária desempenha papéis críticos, como o roteamento inteligente, que direciona o tráfego do domínio principal para a *Vercel* e o subdomínio da API para a infraestrutura na *DigitalOcean*. Adicionalmente, a *Cloudflare* atua como um *proxy* reverso, provendo a terminação SSL necessária para que a comunicação ocorra exclusivamente por meio de protocolos seguros, garantindo a integridade dos dados trafegados entre o usuário e o servidor.

Capítulo 4

Resultados

Neste capítulo, serão apresentados os resultados alcançados com o desenvolvimento do Sistema de Gestão e Monitoramento de Colmeias de Abelhas *Apis*, um *software* criado para melhorar e otimizar a gestão apícola, como também a validação do *hardware* do protótipo desenvolvido. Em seguida, descrevemos as principais funcionalidades do sistema, ilustrando seu funcionamento e destacando os benefícios que proporciona aos usuários.

4.1 Montagem e validação do nó sensor

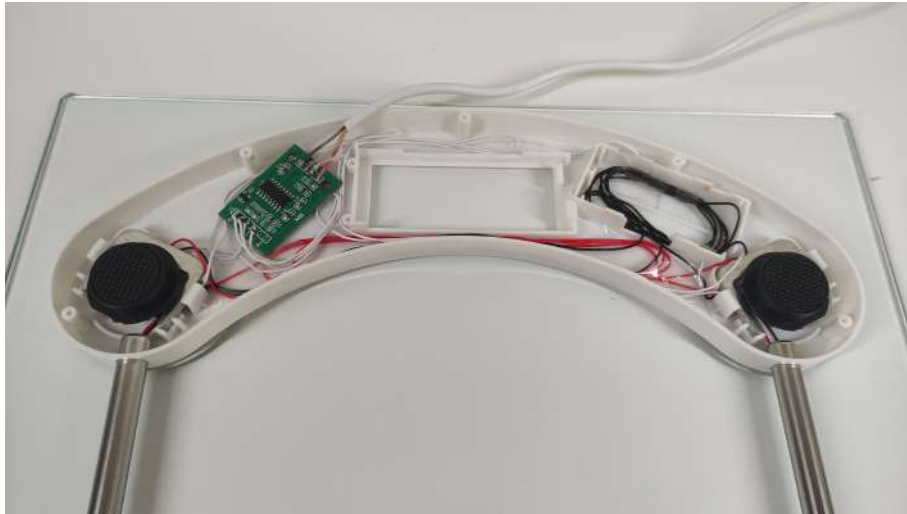
Para a construção do protótipo, foi utilizada uma balança corporal, exposta na Figura 4.1, onde acoplado o módulo amplificador HX711, disposto na figura 4.2, a balança também exerce o papel de base de sustentação da colmeia padrão do projeto, onde fica acoplado o hardware embarcado protegido com material plástico, sensores ambientais e placa solar, conforme representação da Figura 4.3.

Figura 4.1: *Estrutura da balança corporal.*



Fonte: Elaborado pelos autores (2025)

Figura 4.2: *Módulo HX711 acoplado na estrutura.*



Fonte: Elaborado pelos autores (2025)

Figura 4.3: *Protótipo acoplado na colmeia Langstroth.*



Fonte: Elaborado pelos autores (2025)

4.1.1 Validação de Confiabilidade da balança

O sistema de pesagem do protótipo utiliza um conjunto de quatro células de carga de 50 kg cada, totalizando uma capacidade máxima de 200 kg, e o módulo HX711 para condicionamento e leitura do sinal. A validação a seguir quantifica a precisão e a margem de erro da estrutura montada em seu ambiente de operação, comparando suas leituras com uma balança de referência, utilizando-se amostras de aproximadamente 1 kg.

Eficiência

A eficiência do balança é medida pela sua capacidade de fornecer leituras consistentes e precisas dentro de sua faixa operacional.

- Capacidade Nominal: O sistema é dimensionado para suportar até 200 kg, fornecendo uma robusta margem de segurança para o peso máximo esperado da colmeia (50 kg a 70 kg).
- Amostragem de Teste: O teste de eficiência foi realizado com amostras de aproximadamente 1 kg.

O módulo HX711 atua como um conversor analógico-digital (A/D) de alta precisão (24 bits). Sua eficiência está ligada à resolução do sistema, que determina a menor variação de peso detectável. Apesar da alta capacidade (200 kg), a calibração do sistema é fundamental para garantir que a margem de erro seja inferior a 3%, para que não haja grandes inconsistências entre o valor real e o valor mensurado.

Método de ensaio

Para estabelecer a margem de erro, foi realizar a comparação entre as leituras do balança do protótipo devidamente calibrada e as leituras de duas balanças de referência certificadas pelo INMETRO disponíveis supermercados em duas amostras distintas de pacotes de arroz de mesma marca, com peso nominal de 1 kg, onde o arroz pesado no supermercado 1 será denominado "Arroz 1", assim como o pesado no supermercado 2, denominado de "Arroz 2", com resultados dispostos na tabela 4.1.

Tabela 4.1: Resultado das pesagens.

Produto	Referência	Protótipo	Diferença (Erro)	Erro Absoluto
Arroz 1	1004 g	994 g	10 g	1,0%
Arroz 2	992 g	987 g	5g	0,5%

Fonte: Elaborado pelos autores (2025)

Os resultados obtidos evidenciam que o sistema de pesagem desenvolvido apresenta desempenho satisfatório dentro da faixa operacional prevista, com erros absolutos inferiores a 3%, conforme estabelecido como limite aceitável para a calibração. A comparação com balanças de referência certificadas pelo INMETRO demonstrou que, mesmo em condições experimentais simples, o protótipo foi capaz de fornecer leituras consistentes e confiáveis. Dessa forma, a validação confirma a viabilidade da solução proposta para o monitoramento do peso das colmeias, assegurando robustez e precisão suficientes para apoiar análises posteriores e a tomada de decisão por parte dos produtores.

4.1.2 Validação de Confiabilidade e Estabilidade da Telemetria

Para avaliar a robustez do fluxo de telemetria, que compreende desde a coleta do nó sensor até a persistência no banco de dados, realizou-se um ensaio de transmissão contínua em ambiente não controlado. O experimento submeteu o protótipo às oscilações reais de uma rede *Wi-Fi* de uso doméstico, com o intuito de simular condições de instabilidade de sinal.

A coleta de dados para esta análise ocorreu no dia 27 de novembro de 2025, em um ciclo de monitoramento contínuo com duração de aproximadamente nove horas, entre 09:48 e 18:35. Esse intervalo de observação permitiu a caracterização do comportamento do sistema frente às variações de sinal da rede sem fio e a validação do mecanismo de hibernação (*Deep Sleep*), configurado para ciclos de 300 segundos.

Eficiência de Entrega

Com base no intervalo programado, a estimativa de envio foi de 105 pacotes de dados. O sistema registrou o recebimento efetivo de 91 mensagens, o que resulta em uma taxa de entrega de 86,67%. Este percentual decorre da ausência de mecanismos de confirmação de recebimento no nível de serviço básico do protocolo MQTT (QoS 0), somada às possíveis instabilidades de redes sem fio em ambientes residenciais. No contexto específico do monitoramento apícola, no qual variáveis como temperatura e umidade apresentam variações lentas, a perda pontual de pacotes não compromete a integridade do histórico nem a fidelidade das curvas de tendência. Dessa forma, os dados persistidos mostram-se suficientes para a representação fidedigna das condições internas da colmeia durante o período analisado.

Análise Temporal e Comportamental

Conforme apresentado na Figura 4.4, que detalha os registros de telemetria persistidos no banco de dados durante o ensaio, a análise dos *timestamps* de recepção permitiu identificar dois padrões distintos de comportamento do *firmware*:

1. Ciclos Nominais (Sucesso): A maioria das transmissões ocorreu com intervalos entre 303 e 306 segundos. A diferença de 3 a 6 segundos em relação ao *Deep Sleep* programado corresponde ao *overhead* operacional necessário para o *boot* do microcontrolador, inicialização dos periféricos e negociação de conexão *Wi-Fi*.
2. Eventos de Perda (*Gaps*): Foram registrados intervalos ampliados, variando entre 600 segundos (perda de um ciclo) e picos isolados de 900 segundos (perda de dois ciclos consecutivos). Essas ocorrências são atribuídas a instabilidades momentâneas na rota da rede durante o envio, onde o pacote foi transmitido pelo dispositivo mas não alcançou o *broker*.

Figura 4.4: Dados armazenados no banco de dados durante o teste.

sensor_readings 1 X						
SELECT timestamp, weight, internal_temperature, internal_humidity, external_temperature						
Insira uma expressão SQL para filtrar os resultados (use Ctrl+Espaço)						
	timestamp	weight	internal_temperature	internal_humidity	external_temperature	
1	2025-11-27 18:35:32.000 -0300	1,88	25,6	48,31	24,44	
2	2025-11-27 18:30:28.000 -0300	1,882	25,57	48,74	24,5	
3	2025-11-27 18:25:24.000 -0300	1,886	25,63	48,88	24,5	
4	2025-11-27 18:10:10.000 -0300	1,897	25,71	49,71	24,56	
5	2025-11-27 18:05:06.000 -0300	1,904	25,72	49,92	24,56	
6	2025-11-27 18:00:02.000 -0300	1,91	25,69	50,39	24,63	
7	2025-11-27 17:54:59.000 -0300	1,916	25,72	50,76	24,69	
8	2025-11-27 17:49:55.000 -0300	1,923	25,8	51,1	24,81	
9	2025-11-27 17:44:51.000 -0300	1,932	25,89	51,38	24,87	
10	2025-11-27 17:34:43.000 -0300	1,95	25,96	52,11	25	
11	2025-11-27 17:29:40.000 -0300	1,96	25,97	52,7	25,06	
12	2025-11-27 17:24:35.000 -0300	1,969	25,97	53,08	25	
13	2025-11-27 17:19:29.000 -0300	1,982	26,02	53,55	24,94	
14	2025-11-27 17:14:25.000 -0300	1,992	25,92	54,49	25,06	
15	2025-11-27 17:09:22.000 -0300	2,003	26,02	55,75	25,25	
16	2025-11-27 17:04:18.000 -0300	2,01	26,12	56,7	25,37	
17	2025-11-27 16:54:08.000 -0300	2,012	26,24	59,46	25,75	
18	2025-11-27 16:49:03.000 -0300	2,007	26,39	61,53	26,06	
19	2025-11-27 16:43:58.000 -0300	2	26,4	62,09	26,06	
20	2025-11-27 16:33:44.000 -0300	1,988	26,61	60,15	26,75	
21	2025-11-27 16:28:40.000 -0300	1,98	26,86	58,54	27,19	
22	2025-11-27 16:23:37.000 -0300	1,97	26,83	58,12	27,13	
23	2025-11-27 16:18:33.000 -0300	1,961	26,8	57,18	27,06	
24	2025-11-27 16:13:30.000 -0300	1,952	26,79	56,72	27	
25	2025-11-27 16:08:26.000 -0300	1,944	26,79	56,06	26,94	
26	2025-11-27 15:58:18.000 -0300	1,931	26,73	54,25	26,81	
27	2025-11-27 15:53:14.000 -0300	1,925	26,72	53,08	26,69	
28	2025-11-27 15:48:11.000 -0300	1,921	26,68	52,29	26,56	

Fonte: Elaborado pelos autores (2025)

Um aspecto crítico validado pelo teste foi a eficácia do mecanismo de autorrecuperação e do sistema de *buffer* local já implementado. A distribuição esparsa dos *gaps* demonstra que o nó sensor não entrou em estado de falha crítica, uma vez que, em 100% dos casos de instabilidade, o dispositivo retomou o ciclo operacional de forma autônoma.

A estratégia de desconexão rápida em caso de falha também contribui significativamente para a eficiência energética. Considerando que o consumo de pico do ESP32 com *Wi-Fi* ativo é de aproximadamente 200 mA (conforme demonstrado na Seção 3.3), a lógica implementada otimiza a autonomia da bateria sob duas condições: em operação normal, o tempo de atividade (*uptime*) é minimizado para cerca de 4 segundos, mantendo o dispositivo desligado em mais de 98% do período total; e em caso de falha de conectividade, ao detectar a indisponibilidade da rede ou do *broker*, o dispositivo aborta a tentativa de transmissão e retorna imediatamente ao modo *Deep Sleep*. Essa abordagem evita o consumo excessivo de energia em tentativas infrutíferas de reconexão, preservando a carga para ciclos subsequentes.

Vale ressaltar que as perdas pontuais observadas são características da limitação da biblioteca MQTT utilizada no protótipo, que opera em QoS 0. Para a evolução do projeto, sugere-se a substituição por bibliotecas que suportem nativamente QoS 1 ou QoS 2, garantindo a integridade total dos dados trafegados.

4.1.3 Validação do Mecanismo de Tolerância a Falhas (*Buffer Offline*)

Para validar a resiliência da solução em cenários de indisponibilidade de rede, realizou-se um ensaio de desconexão prolongada. O objetivo consistiu em verificar a eficácia do armazenamento local em memória não volátil e analisar o comportamento da sincronização de dados no momento em que a conectividade é restaurada.

A coleta de dados para esta análise compreendeu o período entre as 21:55 do dia 27 de novembro e as 06:46 do dia 28 de novembro de 2025, totalizando aproximadamente dez horas de operação *offline*. Nesse intervalo, o *firmware* identificou a ausência de rede e iniciou automaticamente a persistência das leituras na memória *flash* interna do microcontrolador até que, às 06:51, a conexão foi restabelecida, disparando o processo de descarregamento do *buffer* e a sincronização com a nuvem.

Análise da Sincronização e Fluxo Cadenciado

O processo de sincronização durou 52 minutos para transmitir 106 registros. O *firmware* utiliza a lógica de Envio Sequencial Cadenciado, limitando cada lote a 11 registros (1 leitura atual e 10 do *buffer*) com intervalos de 1 segundo. Essa estratégia assegura a estabilidade da pilha TCP/IP e evita a sobrecarga do *modem*.

Conforme ilustrado na Figura 4.5, os dados chegam em lotes de até 11 mensagens. A eficácia técnica é comprovada pelo contraste temporal: os *timestamps* preservam o horário real da coleta (ocorrida horas antes), enquanto o campo *created_at* atesta a inserção sequencial e acelerada no banco de dados durante a fase de recuperação.

Figura 4.5: Dados armazenados no banco de dados durante o teste.

	hive_id	timestamp	created_at	123 weight	123 internal_temperature	123 internal_humidity
1	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 07:43:26.000 -0300	2025-11-28 07:43:34.692 -0300	2,004	27,28	72,78
2	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 06:46:03.000 -0300	2025-11-28 07:43:33.828 -0300	1,988	27,2	69,56
3	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 06:40:54.000 -0300	2025-11-28 07:43:32.806 -0300	1,988	27,17	69,81
4	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 06:35:45.000 -0300	2025-11-28 07:43:31.778 -0300	1,989	27,17	68,65
5	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 06:30:36.000 -0300	2025-11-28 07:43:30.705 -0300	1,989	27,15	69,34
6	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 07:38:11.000 -0300	2025-11-28 07:38:26.821 -0300	2,003	27,31	71,92
7	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 06:25:27.000 -0300	2025-11-28 07:38:26.012 -0300	1,99	27,12	68,53
8	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 06:20:18.000 -0300	2025-11-28 07:38:24.888 -0300	1,991	27,13	68,98
9	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 06:15:10.000 -0300	2025-11-28 07:38:23.863 -0300	1,994	27,12	68,5
10	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 06:10:01.000 -0300	2025-11-28 07:38:22.651 -0300	1,996	27,15	68,08
11	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 06:04:52.000 -0300	2025-11-28 07:38:21.612 -0300	1,998	27,14	68,11
12	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:59:43.000 -0300	2025-11-28 07:38:20.486 -0300	2,002	27,13	67,69
13	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:54:34.000 -0300	2025-11-28 07:38:19.463 -0300	2,004	27,11	67,48
14	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:49:26.000 -0300	2025-11-28 07:38:18.336 -0300	2,008	27,08	67,71
15	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:44:17.000 -0300	2025-11-28 07:38:17.318 -0300	2,011	27,08	68,28
16	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:39:08.000 -0300	2025-11-28 07:38:16.422 -0300	2,014	27,05	67,98
17	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:33:59.000 -0300	2025-11-28 07:33:11.849 -0300	2,017	27,07	69,67
18	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:28:50.000 -0300	2025-11-28 07:33:10.722 -0300	2,019	27,05	68,28
19	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:23:41.000 -0300	2025-11-28 07:33:09.698 -0300	2,019	27,03	69,25
20	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:18:33.000 -0300	2025-11-28 07:33:08.573 -0300	2,019	27,01	69,4
21	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:13:24.000 -0300	2025-11-28 07:33:07.580 -0300	2,019	27,02	71,06
22	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:08:15.000 -0300	2025-11-28 07:33:06.526 -0300	2,02	27,02	71,26
23	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 05:03:06.000 -0300	2025-11-28 07:33:05.400 -0300	2,018	27,01	70,01
24	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 04:57:57.000 -0300	2025-11-28 07:33:04.378 -0300	2,018	27,02	71,15
25	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 04:52:48.000 -0300	2025-11-28 07:33:03.355 -0300	2,018	27,03	70,6
26	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 04:47:39.000 -0300	2025-11-28 07:33:02.279 -0300	2,016	27,01	68,77
27	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 07:27:43.000 -0300	2025-11-28 07:27:58.591 -0300	1,998	27,37	72,04
28	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 04:42:30.000 -0300	2025-11-28 07:27:57.783 -0300	2,015	27	69,38
29	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 04:37:22.000 -0300	2025-11-28 07:27:56.660 -0300	2,013	27,01	69,21
30	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 04:32:13.000 -0300	2025-11-28 07:27:55.633 -0300	2,011	27	69,02
31	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 04:27:04.000 -0300	2025-11-28 07:27:54.613 -0300	2,01	27,01	69,1
32	96b1a434-29e1-4054-8c40-ee6b42cda9c1	2025-11-28 04:21:55.000 -0300	2025-11-28 07:27:53.484 -0300	2,000	27,01	67,76

Fonte: Elaborado pelos autores (2025)

A análise dos dados persistidos após a sincronização revelou os seguintes comportamentos:

1. **Ciclos de Sucesso Pleno:** Em aproximadamente 70% dos ciclos de sincronização, o sistema obteve sucesso na entrega da carga total de 11 registros. O intervalo de 1 segundo entre envios mostrou-se eficaz para manter a estabilidade da conexão durante a transmissão do lote.
2. **Perdas por Limitação do Protocolo:** Nos demais ciclos, observou-se a chegada de 8 a 10 registros. A análise técnica indica que, embora o envio seja cadenciado, o protocolo MQTT com QoS 0 não fornece garantia de entrega. Em situações de instabilidade na rede, o *firmware* reporta o envio como bem-sucedido ao descarregar o pacote no rádio, o que resulta na exclusão do arquivo local, ainda que o pacote sofra descarte na rota antes de atingir o *broker*.
3. **Conclusão da Fila:** O último ciclo (07:43) registrou a chegada de 5 arquivos, confirmando o esvaziamento completo do *buffer*.

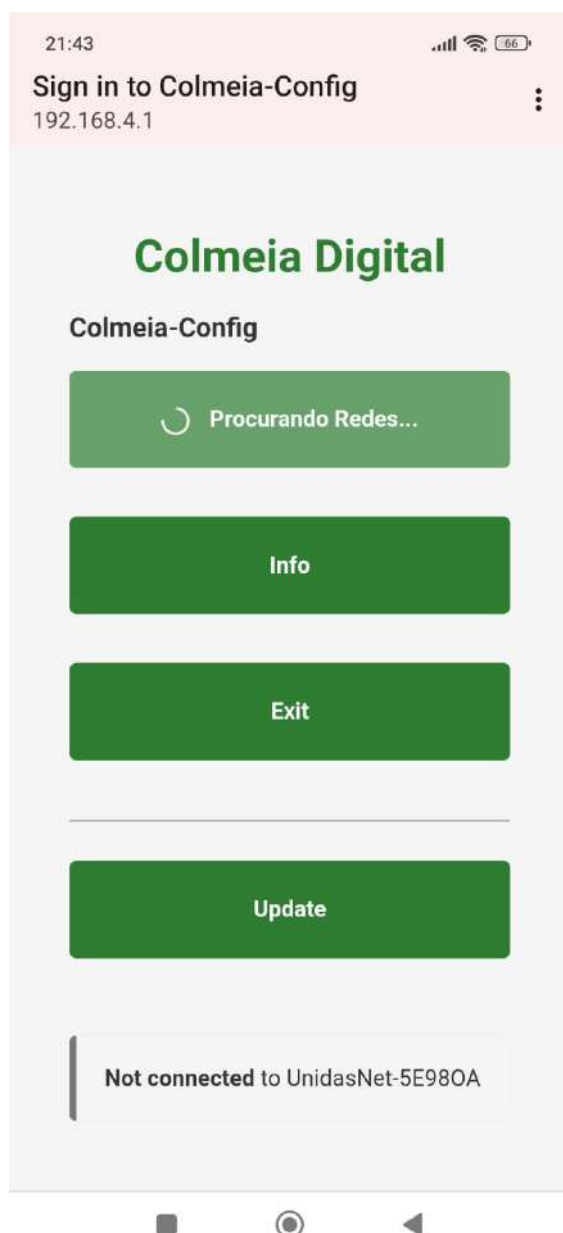
O teste validou a eficácia do mecanismo de *buffer offline* como estratégia para mitigar a perda massiva de informações durante períodos extensos de desconexão. Contudo, os resultados evidenciam que a garantia de entrega total (100%) em redes instáveis requer, para trabalhos futuros, a migração para o nível QoS 1, condicionando a exclusão do dado local à confirmação de recebimento emitida pelo servidor.

4.2 Apresentação da plataforma Web

4.2.1 Inicialização e Configuração de Conectividade

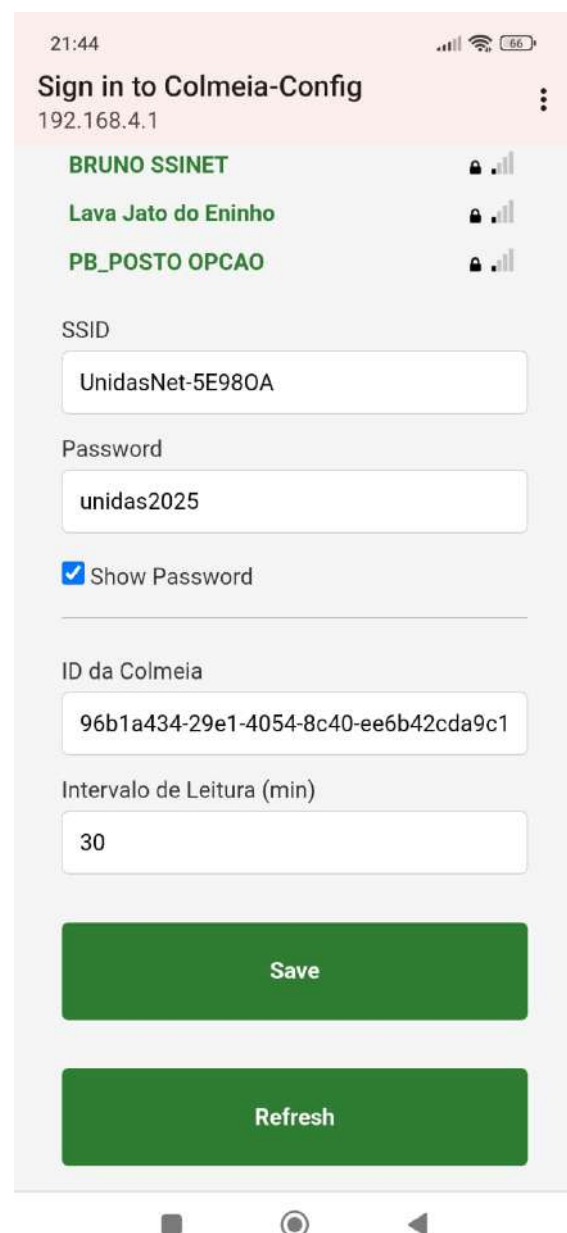
O provisionamento do nó sensor é realizado via interface *web* (*Captive Portal*), acessível automaticamente na primeira inicialização ou manualmente, mediante o acionamento do botão físico por mais de 6 segundos. Em ambos os cenários, o usuário conecta-se ao Ponto de Acesso (AP) gerado pelo dispositivo para definir os parâmetros operacionais essenciais: as credenciais da rede *Wi-Fi*, a identificação da colmeia e o intervalo entre ciclos de leitura, conforme ilustrado nas Figuras 4.6 e 4.7.

Figura 4.6: Tela captive portal.



Fonte: Elaborado pelos autores (2025)

Figura 4.7: Tela de Configuração.

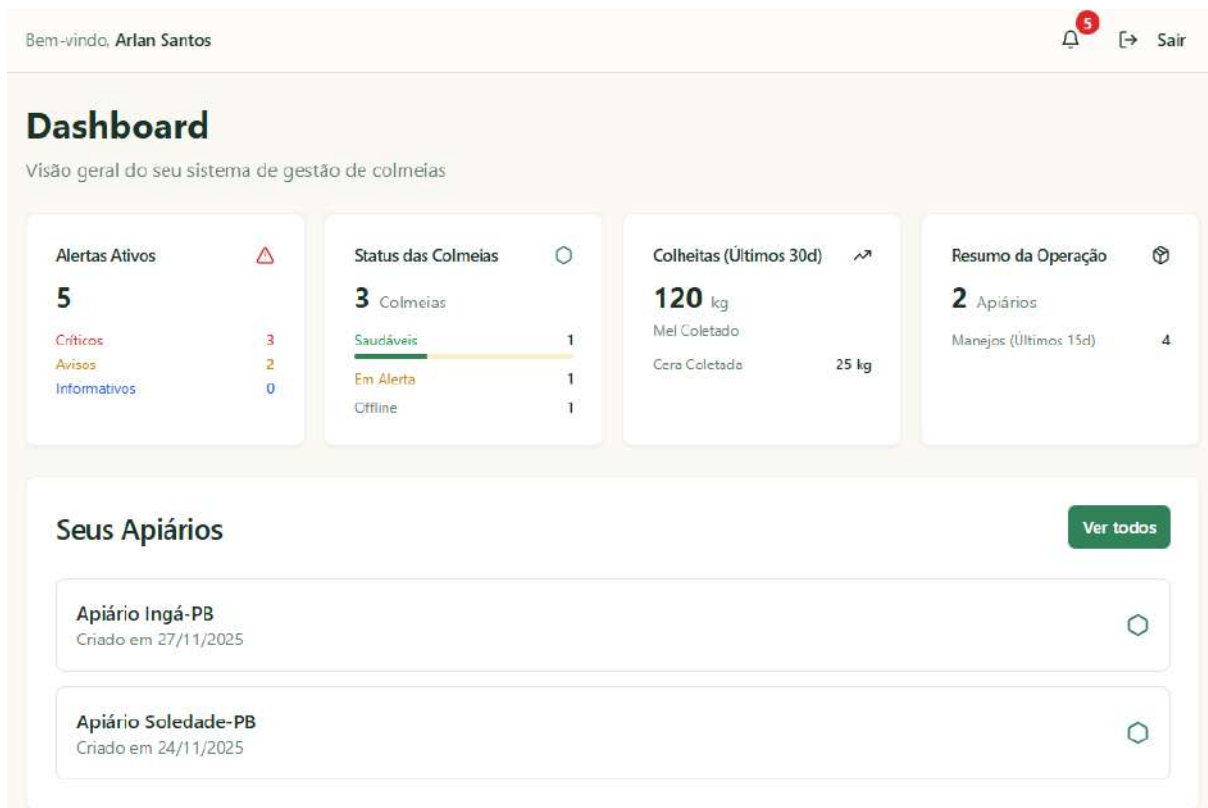


Fonte: Elaborado pelos autores (2025)

4.2.2 Dashboard

A interface inicial do sistema trata-se de um *Dashboard* que reúne as principais informações sobre as colmeias, tais como: alertas, status, colheitas, resumo e listagem dos apiários. Esta interface centraliza as informações críticas para a tomada de decisão rápida, conforme ilustrado na Figura 4.8.

Figura 4.8: Tela de dashboard.



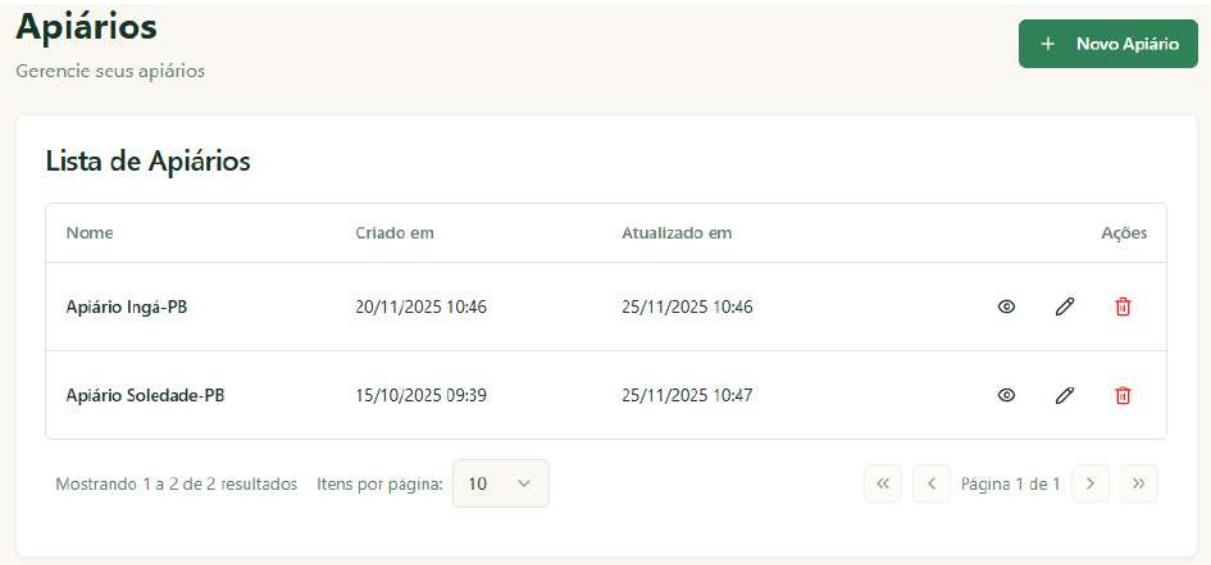
Fonte: Elaborado pelos autores (2025)

4.2.3 Apiários

A organização lógica do sistema inicia-se pela definição dos Apiários, que funcionam como *contêineres* para agrupar as colmeias geograficamente.

A Figura 4.9 ilustra a interface de gerenciamento de apiários, desenvolvida para atender integralmente ao requisito funcional (RF11), listando os apiários existentes e permitindo as operações de criação, leitura, atualização e remoção.

Figura 4.9: Tela da listagem de apiários.

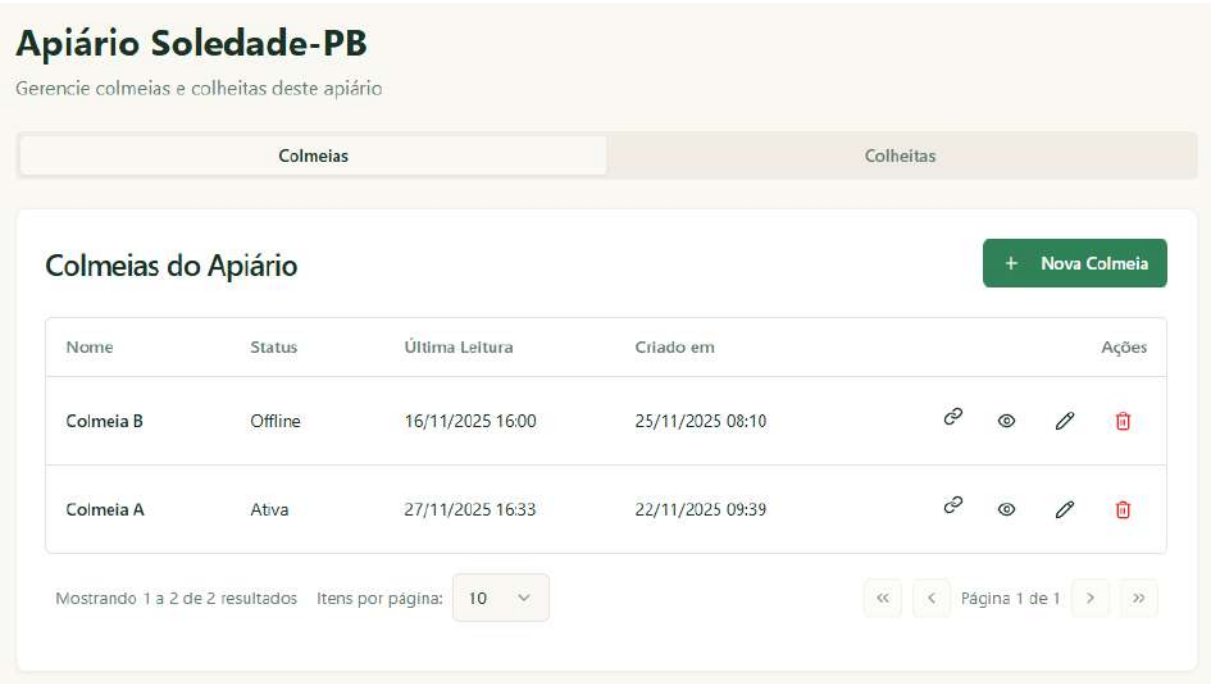


Fonte: Elaborado pelos autores (2025)

Colmeias do apiário

Ao selecionar um apiário na listagem principal, o usuário é direcionado para uma visão detalhada que centraliza os ativos vinculados à unidade selecionada, conforme ilustrado na Figura 4.10. A aba principal desta interface, *Colmeias*, é responsável pelo gerenciamento de parte do ciclo de vida das caixas de abelhas, atendendo ao Requisito Funcional (RF12).

Figura 4.10: Tela da listagem de colmeias do apiário.



Fonte: Elaborado pelos autores (2025)

Colheitas do apiário

Ainda dentro da visão detalhada do apiário, a plataforma oferece um módulo específico para o controle de produção, acessível através da aba *Colheitas*. Esta funcionalidade atende ao Requisito Funcional (RF16), digitalizando os registros de campo e gerando métricas históricas de desempenho, conforme ilustrado nas Figuras 4.11 e 4.12, que apresentam, respectivamente, as interfaces de consulta e o gerenciamento de dados.

Figura 4.11: Tela da listagem de colheitas.

Apiário Soledade-PB

Gerencie colmeias e colheitas deste apiário

ColmeiasColheitas

Colheitas do Apiário

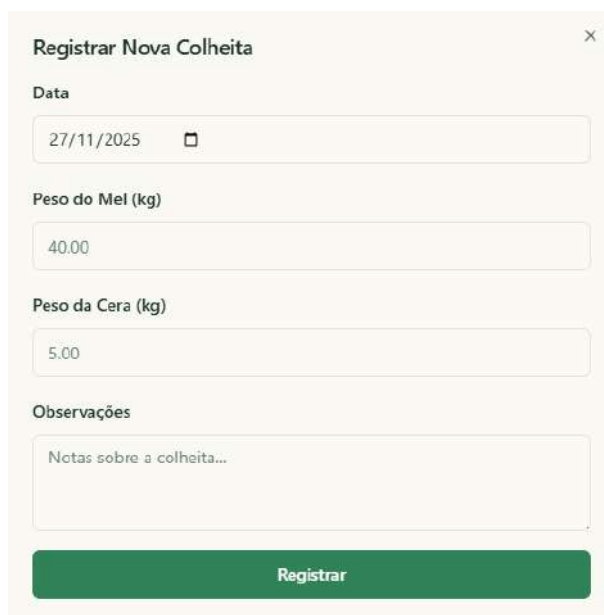
+ Nova Colheita

Data	Mel (kg)	Cera (kg)	Notas	Ações
27/11/2025	40,00 kg	15,00 kg	-	 
20/10/2025	35,00 kg	10,00 kg	6 Horas de Colheita	 

Mostrando 1 a 2 de 2 resultados Itens por página: 10 

  Página 1 de 1  

Fonte: Elaborado pelos autores (2025)

Figura 4.12: Tela de registro de colheita.A imagem mostra uma interface web para registrar uma nova colheita. O formulário é intitulado "Registrar Nova Colheita" e possui um ícone de fechar (X) no canto superior direito. Ele contém os seguintes campos: "Data" com o valor "27/11/2025" e um ícone de calendário; "Peso do Mel (kg)" com o valor "40.00"; "Peso da Cera (kg)" com o valor "5.00"; e "Observações" com o texto "Notas sobre a colheita...". Um botão verde "Registrar" está localizado na base do formulário.

Fonte: Elaborado pelos autores (2025)

4.2.4 Gerenciamento de alertas

Para garantir que anomalias detectadas nas colmeias (como perda súbita de peso ou temperaturas fora da faixa ideal) sejam tratadas com agilidade, foi implementado um módulo de alertas em tempo real, atendendo ao requisito (RF18). A interface foi projetada para suportar um fluxo de trabalho completo de resolução de incidentes, dividido em notificação rápida (Figura 4.13) e gerenciamento detalhado.

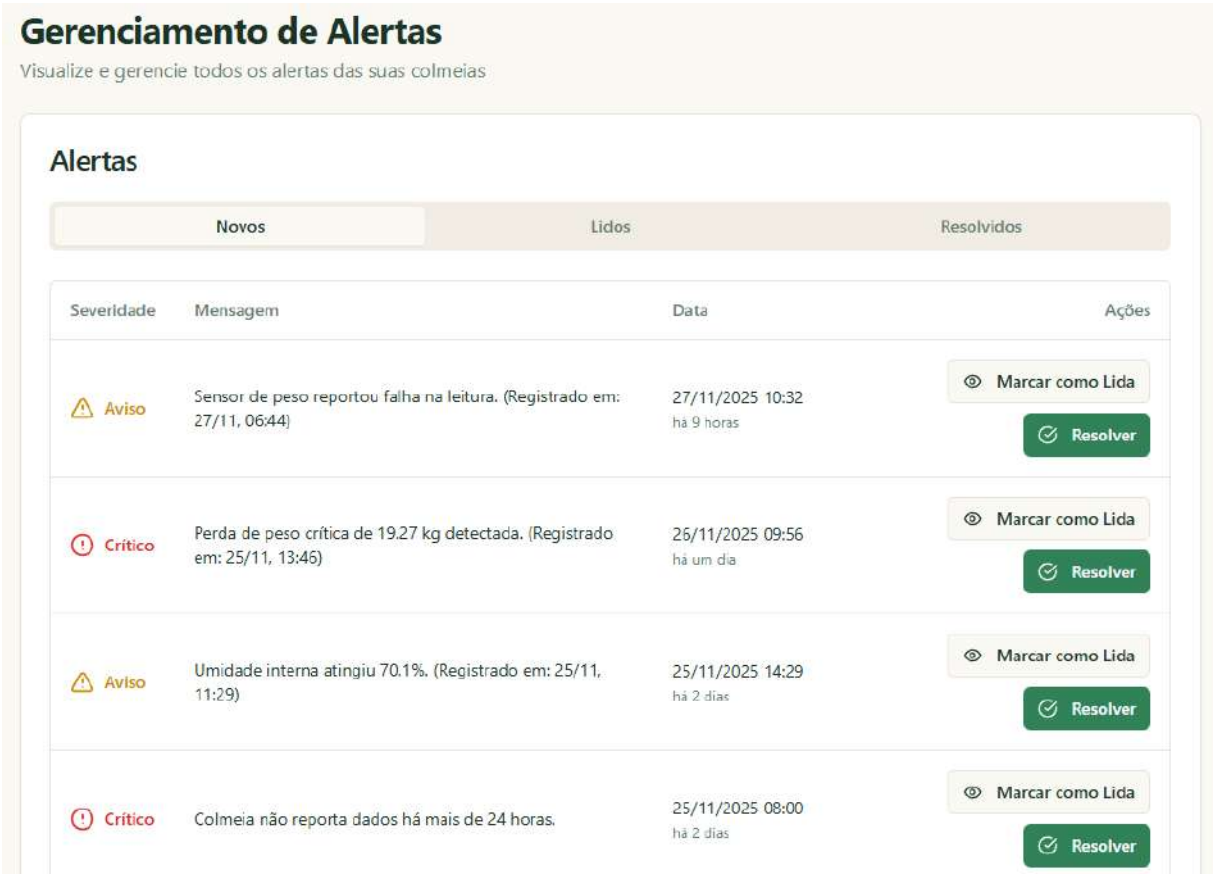
Na interface de *Gerenciamento de Alertas* é possível acessar cada um dos alertas das colmeias. O usuário pode selecionar entre três categorias de alertas: Novos (Figura 4.14), Lidos (Figura 4.15) e Resolvidos (Figura 4.16). Dessa forma, o sistema permite gerir os alertas por meio da ferramenta de marcação, possibilitando que sejam classificados como lidos ou resolvidos de maneira prática e organizada.

Figura 4.13: Notificações de alertas.



Fonte: Elaborado pelos autores (2025)

Figura 4.14: Tela de exibição de alertas novos.



Fonte: Elaborado pelos autores (2025)

Figura 4.15: Tela de exibição de alertas lidos.

Gerenciamento de Alertas

Visualize e gerencie todos os alertas das suas colmeias

Alertas

Novos

Lidos

Resolvidos

Severidade	Mensagem	Data	Ações
 Aviso	Sensor de peso reportou falha na leitura. (Registrado em: 27/11, 06:44)	27/11/2025 10:32 há 9 horas	 Resolver
 Critico	Perda de peso crítica de 19.27 kg detectada. (Registrado em: 25/11, 13:46)	26/11/2025 09:56 há um dia	 Resolver
 Aviso	Umidade interna atingiu 70.1%. (Registrado em: 25/11, 11:29)	25/11/2025 14:29 há 2 dias	 Resolver
 Critico	Colmeia não reporta dados há mais de 24 horas.	25/11/2025 08:00 há 2 dias	 Resolver
 Critico	Temperatura interna caiu para 26.95°C. (Registrado em: 24/11, 06:41).	24/11/2025 09:41 há 3 dias	 Resolver

Mostrando 1 a 5 de 5 resultados

Itens por página: 10

<<

<

Página 1 de 1

>

>>

Fonte: Elaborado pelos autores (2025)

Figura 4.16: Tela de exibição de alertas resolvidos.

Gerenciamento de Alertas
Visualize e gerencie todos os alertas das suas colmeias

Alertas

Novos	Lidos	Resolvidos
-------	-------	------------

Severidade	Mensagem	Data	Ações
 Aviso	Sensor de peso reportou falha na leitura. (Registrado em: 27/11, 06:44)	27/11/2025 10:32 há 9 horas	
 Aviso	Umidade interna atingiu 70.1%. (Registrado em: 25/11, 11:29)	25/11/2025 14:29 há 2 dias	
 Crítico	Colmeia não reporta dados há mais de 24 horas.	25/11/2025 08:00 há 2 dias	

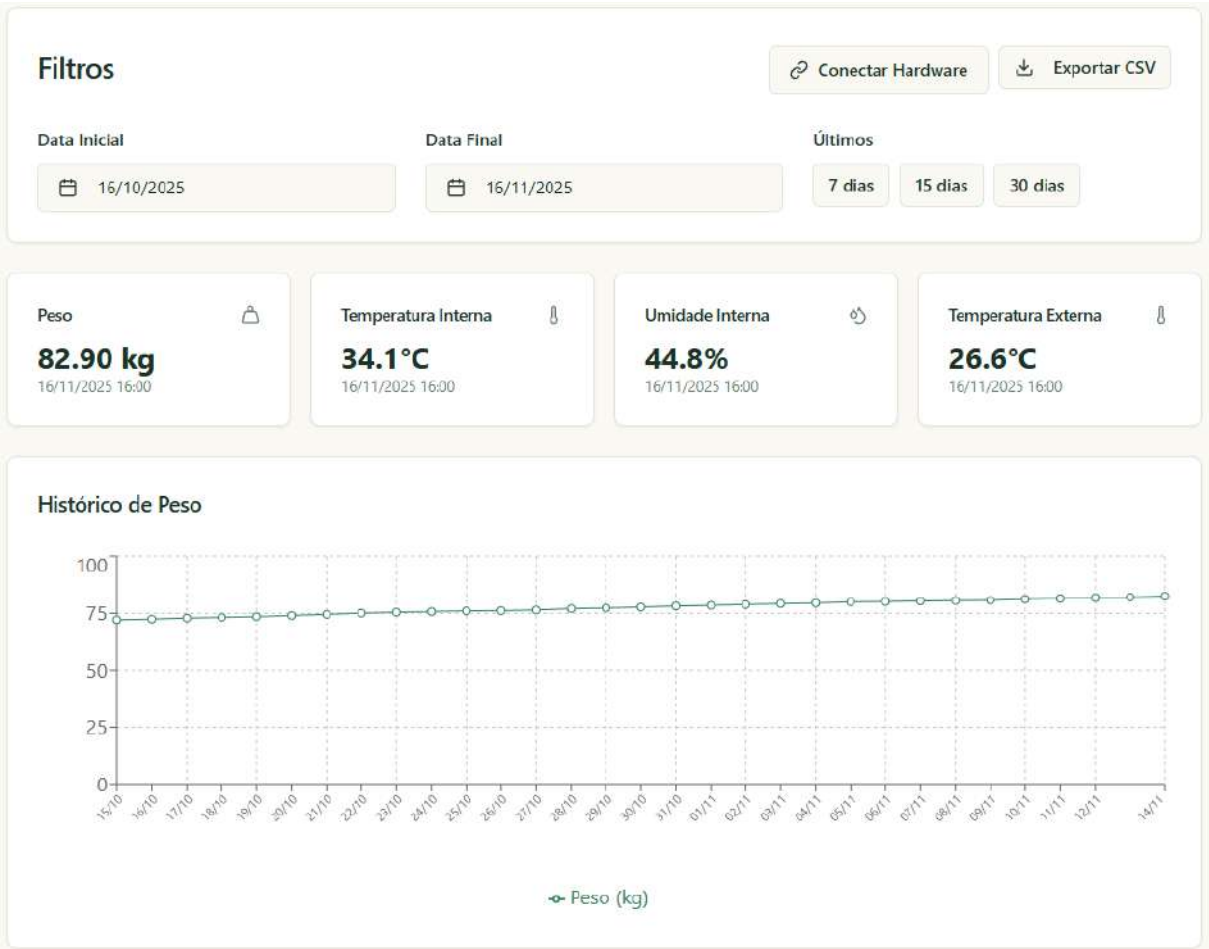
Mostrando 1 a 3 de 3 resultados Itens por página: 10 << < Página 1 de 1 > >>

Fonte: Elaborado pelos autores (2025)

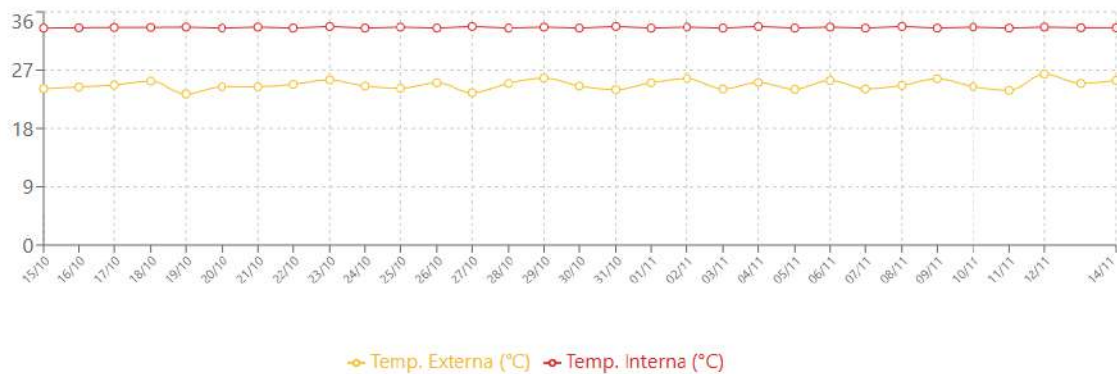
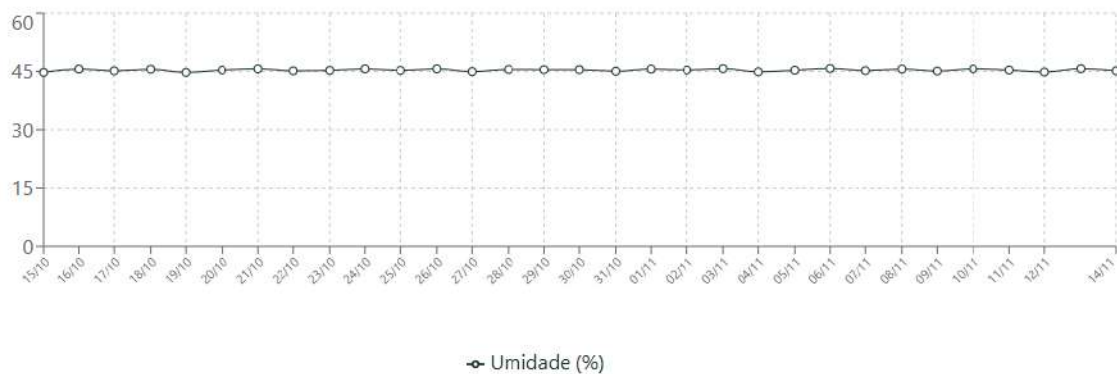
4.2.5 Histórico da colmeia

A tela de *Histórico da colmeia* apresenta os registros periódicos dos sensores da colmeia selecionada, além da exportação dos dados no formato de arquivo CSV, ilustrada nas Figuras 4.17 e 4.18, que permite a correlação entre variáveis ambientais e físicas, essencial para o diagnóstico de saúde da colônia, que atende ao Requisito Funcional (RF15).

Figura 4.17: Painel de sensoriamento da colmeia



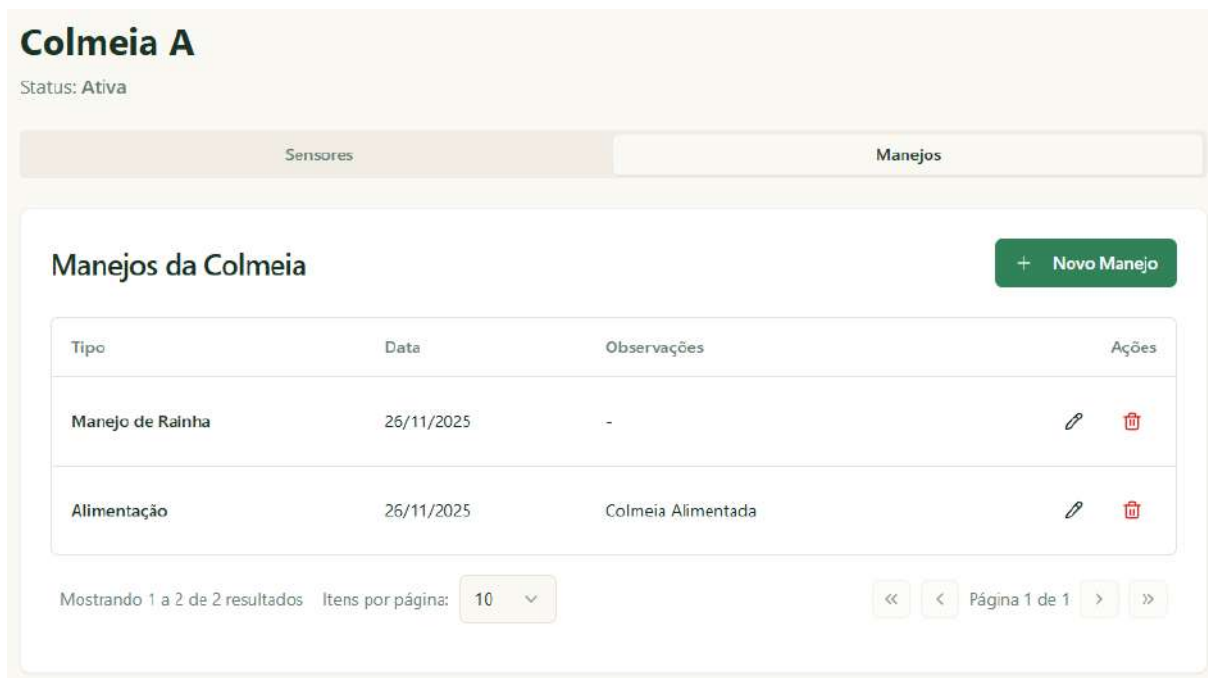
Fonte: Elaborado pelos autores (2025)

Figura 4.18: *Gráficos de temperatura e umidade***Histórico de Temperatura****Histórico de Umidade**

Fonte: Elaborado pelos autores (2025)

Manejos da colmeia

Complementando o processo de monitoramento, o sistema oferece uma interface para o registro das intervenções humanas realizadas na colmeia, atendendo ao Requisito Funcional (RF16). Esta funcionalidade atua como um *diário de campo digital*, essencial para correlacionar as variações observadas nos sensores com as ações de manutenção, conforme ilustrado na Figura 4.19. Ao alternar para a aba de manejos, é exibida a lista de manejos realizados, seus tipos e suas respectivas datas, além das ações de inclusão, anotação, edição e remoção de cada registro.

Figura 4.19: Tela de manejos da colmeia

The screenshot displays the 'Colmeia A' management interface. At the top, the title 'Colmeia A' is followed by 'Status: Ativa'. Below this, there are two tabs: 'Sensores' and 'Manejos', with 'Manejos' being the active tab. The main section is titled 'Manejos da Colmeia' and includes a '+ Novo Manejo' button. A table lists the treatments with columns for 'Tipo', 'Data', 'Observações', and 'Ações'. Two entries are shown: 'Manejo de Rainha' and 'Alimentação', both dated 26/11/2025. The bottom of the interface shows pagination information: 'Mostrando 1 a 2 de 2 resultados', 'Itens por página: 10', and 'Página 1 de 1'.

Tipo	Data	Observações	Ações
Manejo de Rainha	26/11/2025	-	 
Alimentação	26/11/2025	Colmeia Alimentada	 

Fonte: Elaborado pelos autores (2025)

Exportação dos dados

Visando garantir a portabilidade das informações e maior flexibilidade aos usuários, foi implementada a exportação de séries históricas em arquivo no formato CSV. Na interface de histórico da colmeia, ao selecionar a opção de exportar o arquivo CSV, o *download* é realizado e o arquivo já está pronto para uso, nesse caso foi utilizado o *Google sheets* para acessar o arquivo CSV. A validação deste recurso, apresentada pela Figura 4.20, confirma a integridade estrutural dos dados após a importação em planilhas externas. Esta funcionalidade permite o pós-processamento das informações, habilitando a criação de relatórios zootécnicos personalizados e auditorias fora do ambiente da plataforma, conforme demonstrado pela geração de gráficos analíticos externos.

Figura 4.20: Exibição dos dados no Google sheets

Historico do Apiario de Soledade

Arquivo Editar Ver Inserir Formatar Dados Ferramentas Extensões Ajuda Gemini

100%

R\$ % .0 .00 123

Padrã...

10

B I A

	A	B	C	D	E
1	Data/Hora	Peso (kg)	Temp. Interna (°C)	Umidade Interna (%)	Temp. Externa (°C)
2	2025-11-25T14:00:00.000Z	2.204	23.11	66.97	22.69
3	2025-11-26T08:00:00.000Z	2.011	26.75	62.73	26.97
4	2025-11-26T10:00:00.000Z	2.009	28.08	58.96	27.98
5	2025-11-26T12:00:00.000Z	2.060	28.15	57.80	29.15
6	2025-11-26T22:00:00.000Z	1.998	27.44	68.93	27.83

Fonte: Elaborado pelos autores (2025)

Capítulo 5

Conclusão e trabalhos futuros

5.1 Conclusão

A arquitetura implementada permitiu a coleta de dados ambientais e de peso por meio de um nó sensor baseado no microcontrolador ESP32, com a subsequente transmissão e armazenamento em uma infraestrutura de nuvem composta por protocolos de mensageria e banco de dados temporais. A visualização das informações por meio da interface *web* desenvolvida demonstrou-se funcional, permitindo ao usuário o acompanhamento remoto dos apiários em tempo real.

Durante os testes realizados, observou-se que o sistema apresentou uma autonomia energética satisfatória, validando-se a eficácia do gerenciamento de energia e do uso de baterias para a manutenção da operação contínua do *hardware* embarcado. Embora o processo de desenvolvimento tenha exigido a superação de desafios técnicos e ajustes de escopo, a estabilidade da telemetria via Wi-Fi, aliada à capacidade de processamento do *backend*, evidenciou que soluções de baixo custo podem oferecer ferramentas de gestão robustas para o produtor rural.

Dessa forma, diante dos resultados obtidos na validação de todas as camadas do projeto, abrangendo desde o sensoriamento até a interface do usuário, conclui-se que o objetivo geral foi plenamente alcançado, comprovando a viabilidade técnica da solução proposta para a modernização do manejo apícola.

5.2 Limitações do Trabalho

Apesar do êxito observado na execução do projeto, verificaram-se limitações técnicas que delimitaram o escopo da versão atual do sistema. Inicialmente, ressalta-se que a transmissão de dados ficou restrita à cobertura de redes *Wi-Fi* locais, uma vez que a implementação da tecnologia *LoRa*, prevista para ampliar o alcance em zonas rurais remotas, mostrou-se inviável dentro das condições de execução deste ciclo de desenvolvimento. No que concerne à instrumentação de pesagem, observou-se que as células de carga apresentam sensibilidade a flutuações térmicas ambientais, o que pode resultar em pequenas oscilações nos registros. No entanto, tal comportamento não compromete a utilidade da solução, visto que a precisão obtida mantém-se dentro de margens de erro

aceitáveis, permitindo a identificação confiável de tendências e variações de peso nas colmeias. Adicionalmente, por tratar-se de um protótipo fundamentado em placas de desenvolvimento (*Dev-Kits*), a arquitetura física priorizou a agilidade na validação funcional e a modularidade dos componentes. Embora essa configuração tenha cumprido os requisitos de prova de conceito, a transição para uma placa de circuito impresso dedicada e um encapsulamento industrializado representa o passo seguinte para garantir uma integração ainda mais robusta e compacta do conjunto eletrônico.

5.3 Trabalhos futuros

O trabalho desenvolvido abre inúmeras possibilidades de aprimoramento e desenvolvimento futuros. Algumas das principais principais agregações incluem:

- **Inclusão da tecnologia LoRa:** Implementar a comunicação LoRa (*Long Range*) para ampliar significativamente o alcance da rede, possibilitando o monitoramento de apiários em áreas rurais remotas ou sem acesso à infraestrutura Wi-Fi.
- **Montagem em placa PCB:** Migrar o protótipo para uma Placa de Circuito Impresso (PCB) personalizada, utilizando exclusivamente o módulo ESP32 (*WROOM*) ao invés do *DevKit*, garantindo maior compacidade, robustez e confiabilidade do sistema.
- **Inclusão de novos sensores:** Avaliar a integração de sensores acústicos e de vibração, permitindo identificar padrões de comportamento da colônia de forma não invasiva e ampliar a precisão no diagnóstico das condições internas da colmeia.
- **Desenvolvimento de aplicativo mobile:** Criar um aplicativo nativo (iOS/Android) ou híbrido para fornecer alertas instantâneos (*Push Notifications*) sobre o estado das colmeias, permitindo que o produtor acompanhe os dados diretamente pelo celular, sem depender de navegadores *web*.
- **Visualização aprimorada e mapas interativos:** Desenvolver um *dashboard* baseado em Sistema de Informação Geográfica (*Geographic Information System* (GIS)), exibindo a localização das colmeias em mapas interativos e correlacionando dados de monitoramento com variáveis ambientais, como altitude, proximidade de lavouras pulverizadas e condições climáticas locais.
- **Incorporação de recursos de análise preditiva:** Utilizar dados históricos (peso, temperatura, umidade) para treinar modelos de *Machine Learning* (ML), que sejam capazes de prever eventos críticos, antecipar riscos e reduzir perdas na produção.
- **Adaptação para meliponicultura:** Expandir o sistema para monitorar colmeias de abelhas nativas sem ferrão, considerando suas necessidades ambientais específicas e parâmetros de saúde distintos das *Apis mellifera*.

Dessa forma, as possibilidades aqui delineadas não apenas ampliam o escopo tecnológico do protótipo, mas também consolidam sua relevância científica e prática para o setor apícola. A integração de novas ferramentas e recurso reforça o caráter interdisciplinar da projeto, posicionando o sistema como uma base sólida para investigações posteriores e para o desenvolvimento de soluções inovadoras em Agricultura 4.0. Assim, o projeto contribui para o avanço da agricultura digital e para a promoção de práticas sustentáveis, alinhando-se às demandas contemporâneas de eficiência, produtividade e preservação ambiental.

Referências Bibliográficas

- ALFA COMP. **Faixas de operação da comunicação LoRaWan**. s.d. Online documentation. Acesso em: 23. nov 2025. Available on: <<https://www.alfacomp.net/faixas-de-operacao-da-comunicacao-lorawan>>. 31
- ANTUNES, J. M. Polinização pode aumentar em 17% a produção de canola. 2024. Acesso em: 31 out. 2025. Available on: <<https://www.embrapa.br/busca-de-noticias/-/noticia/2631608/polinizacao-pode-aumentar-em-17-a-producao-de-canola>>. 8
- APACAME. **LORENZO LORRAINE LANGSTROTH**. 2022. Mensagem doce. Acesso em: 6 out. 2025. Available on: <<https://apacame.org.br/site/revista/mensagem-doce-n-167-julho-de-2022/curiosidade/>>. 6
- ARDUCORE. **Esp32s**. 2025. Acesso em: 19 set. 2025. Available on: <https://www.arducore.com.br/chip-sem-modulo-esp32s-modelo-s-compativel-c-antena-wifi?srsId=AfmBOor3KjyKRNTUeR71YIca0msmGw_Vx0dtJDAB0jVNrOC0J9k76ILF>. 15
- ARSAIR. **Sensor AHT10 Documentation**. 2018. Online documentation. Acesso em: 17. ago. Available on: <<https://altronics.cl/uploads/AHT10.pdf>>. 27
- AVIA SEMICONDUCTOR. **HX711 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales**. 2018. Datasheet. Acesso em: 25 ago. 2025. Available on: <https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf>. 26
- CASTAGNINO, D. A. D. S. P. G. G. L. B. Produtividade de mel e comportamento defensivo como índices de melhoramento genético de abelhas africanizadas (*Apis mellifera* L.). 2012. Acesso em: 03 set. 2025. Available on: <<https://repositorio.ufba.br/bitstream/ri/7118/1/cc.pdf>>. 8
- CIRAM. **Apis On-line**. 2025. Acesso em: 16 dez. 2025. Available on: <<https://ciram.epagri.sc.gov.br/apicultura/monitor-colmeias.jsp>>. 1
- CLOUDFLARE. **O que é SSL? | SSL vs. TLS**. 2025. Acesso em: 15 dez. 2025. Available on: <<https://www.cloudflare.com/pt-br/learning/ssl/what-is-ssl/>>. 14
- DALLAS SEMICONDUCTOR. **DS18B20 Programmable Resolution 1-Wire Digital Thermometer**. 2018. Online documentation. Acesso em: 15 ago. 2025. Available on: <<https://cdn.sparkfun.com/datasheets/Sensors/Temp/DS18B20.pdf>>. 27
- EMBARCADOS. **Trazendo o mundo real para dentro do processador – Conversor A/D**. 2015. Acesso em: 28 set. 2025. Available on: <<https://embarcados.com.br/conversor-a-d/>>. 18
- EMBARCADOS. **ESP32-S3-DevKitC-1 v1.1**. 2023. Acesso em: 03 set. 2025. Available on: <<https://embarcados.com.br/placa/esp32-s3-devkitc-1-v1-1/>>. 25

EMBRAPA. Pesquisa da embrapa insere abelhas em cafezais e produtividade sobe 16,52024. Acesso em: 31 out. 2025. Available on: <<https://agenciagov.ebc.com.br/noticias/202410/pesquisa-da-embrapa-insere-abelhas-em-cafezais-e-produtividade-sobe-16-5#:~:text=Pesquisa%20da%20Embrapa%20insere%20abelhas,16%2C5%25%20%E2%80%94%20Ag%C3%A2ncia%20Gov>>. 8

ENCYCLOPEDIA BRITANNICA. **Honeybee**. 2025. Acesso em: 08 set. 2025. Available on: <<https://www.britannica.com/animal/honeybee>>. 5

ERDMANN, A. C.; GAVIOLI, A.; FILHO, P. L. D. P.; COLTRO1, Y. C. **Simulação de viabilidade de um sistema de telemetria em caminhões de coleta de laticínios com o uso de uma rede LoraWAN**. 2023. Acesso em: 22 dez. 2025. Available on: <https://www.researchgate.net/publication/370241937_Simulacao_de_viabilidade_de_um_sistema_de_telemetria_em_caminhoes_de_coleta_de_laticinios_com_o_uso_de_uma_rede_LoraWAN>. 14

ESPRESSIF SYSTEMS. **ESP32 Series Technical Reference Manuals and Datasheets**. 2024. Technical documentation portal. Acesso em: 15 ago. 2025. Available on: <<https://www.espressif.com/en/support/documents/technical-documents>>. 16

ESPRESSIF SYSTEMS. **Over The Air Updates (OTA)**. 2024. Acesso em: 17 dez. 2025. Available on: <<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html>>. 19

FORTINET. **O que é firmware? Tipos e exemplos**. 2024. Acesso em: 22 dez. 2025. Available on: <<https://www.fortinet.com/br/resources/cyberglossary/what-is-firmware>>. 19

GAZZONI, D. L. Crise de mão de obra no campo: causas, impactos e possíveis soluções. 2018. Acesso em: 3 set. 2025. Available on: <<https://openknowledge.fao.org/server/api/core/bitstreams/1de25fa0-6792-4c2b-8e93-3c8aa521f08c/content>>. 10

GOOGLE DEVELOPERS. **Por que o HTTPS é importante**. 2020. Acesso em: 16 dez. 2025. Available on: <<https://developers.google.com/search/docs/advanced/security/https?hl=pt-br>>. 14

IBGE. **Censo 2022: 87% da população brasileira vive em áreas urbanas**. 2024. Acesso em: 29 jun. 2025. Available on: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/41901-censo-2022-87-da-populacao-brasileira-vive-em-areas-urbanas>>. 1

JONG, D. D. Africanized honey bees in brazil, forty years of adaptation and success. **Annual Review of Entomology**, Annual Reviews, v. 41, n. 1, p. 555–578, 1996. Acesso em: 10 jul. 2025. Available on: <https://www.researchgate.net/publication/240827833_Africanized_honey_bees_in_Brazil_forty_years_of_adaptation_and_success>. 5, 6, 9

KUBOTA, L. C. Adoção da internet das coisas (iot) na agropecuária: uma revisão sistemática sobre as possibilidades de adoção no ambiente produtivo rural brasileiro. 2023. Acesso em: 4 set. 2025. Available on: <<https://www.interacoes.ucdb.br/interacoes/article/view/4024/3044>>. 11

LIMA, A. O. N. **Produção de própolis por abelha melífera africanizada (*Apis mellifera* L.) na caatinga do baixo jaguaribe cearense**. Tese (Doutorado) — UFC, 2012. Acesso em: 29 out. 2025. Available on: <https://repositorio.ufc.br/bitstream/riufc/17003/1/2012_tese_aonlima.pdf>. 8

LOCKHAT, R. **Physics: Wheatstone bridge**. 2020. Southern African Journal of Anaesthesia and Analgesia. Acesso em: 16 ago. 2025. Available on: <https://www.researchgate.net/publication/348512139_Physics_Wheatstone_bridge>. 18

MACHADO, P. A. Utilização de tecnologia de precisão na apicultura: Uma revisão sistematica. 2023. Acesso em: 4 set. 2025. Available on: <<https://lume.ufrgs.br/bitstream/handle/10183/276197/001198570.pdf?sequence=1>>. 11

MAKERHERO. **O que é um strain gauge?** 2025. Acesso em: 13 nov. 2025. Available on: <https://www.makerhero.com/blog/o-que-e-strain-gauge-arduino/?srsltid=AfmBOopn9GDIODNMBjJ52zWws2o0VALiKGGFpNiz-yxDkayVh_Y1m1WX>. 17

MAKERHERO. **Real Time Clock RTC DS3231**. 2025. Acesso em: 14 set. 2025. Available on: <<https://www.makerhero.com/wp-content/uploads/2017/07/4MD09-1.jpg>>. 26

MALAFAIA, G. C. Crise de mão de obra no campo: causas, impactos e possíveis soluções. 2024. Acesso em: 3 set. 2025. Available on: <<https://www.embrapa.br/documents/1355108/51748908/Boletim+CiCarne+65-2024.pdf>>. 10

MAPA. **Exportações brasileiras de mel natural**. 2021. Acesso em: 15 set. 2025. Available on: <<https://www.semadesc.ms.gov.br/wp-content/uploads/2024/01/Exportacoes-Brasileiras-de-Mel-Natural.pdf>>. 9

MICHELS, C. Z. T. S. A. Desenvolvimento de um equipamento para a produção de cera alveolada. **HOLOS**, IFRN, 2016. Acesso em: 27 nov. 2025. Available on: <<https://www.redalyc.org/pdf/4815/481554865006.pdf>>. 7

MOJARAVSCKI, D. F. **Apicultura Digital**. Tese (Doutorado) — UFSM, 2018. Acesso em: 6 dez. 2025. Available on: <https://repositorio.ufsm.br/bitstream/handle/1/20162/DIS_PPGAP_2018_MOJARAVSCKI_DAVID.pdf?sequence=1&isAllowed=y>. 1

MOZILLA DEVELOPER NETWORK. **Uma visão geral do HTTP**. 2025. Acesso em: 15 dez. 2025. Available on: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>>. 13

OMEGA ENGINEERING. **Load Cells and Force Sensors — Technical Reference**. 2021. Technical Reference. Acesso em: 16 ago. 2025. Available on: <<https://www.omega.com/en-us/resources/load-cells>>. 18

ORELLANA. **Modulo sensor HX711**. 2025. Acesso em: 14 set. 2025. Available on: <<https://orellanaelec.com/producto/modulo-sensor-de-presion-hx711/>>. 26

PRASTYO, E. A. **Sensor Suhu DS18B20**. 2018. Edukasi elektronika. Acesso em: 14 set. 2025. Available on: <<https://www.edukasiElektronika.com/2020/09/sensor-suhu-ds18b20.html>>. 27

ROUBIK, D. W. **Pollination of cultivated plants in the tropics**. FAO, 1995. Acesso em: 21 mai. 2025. Available on: <<https://openknowledge.fao.org/server/api/core/bitstreams/1de25fa0-6792-4c2b-8e93-3c8aa521f08c/content>>. 4

SANTOS, B. P. et al. Internet das coisas: da teoria à prática. In: . SBC, 2016. Acesso em: 17 dez. 2025. Available on: <<https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>>. 10, 11

SILVEIRA, D. C. D.; MARACAJÁ, P. B.; SILVA, R. A. D.; SOUSA, R. M.; BLANCO, B. S. Polinização pode aumentar em 172024. Acesso em: 27 nov. 2025. Available on: <<https://www.scielo.br/j/rbspa/a/Wqg5PPVfxsXMNF4btZZqTWd/?format=pdf&lang=pt>>. 7

Apêndices

Apêndice A

Links dos Repositórios e Aplicação

Neste apêndice, são apresentados os endereços eletrônicos para acesso ao código-fonte completo desenvolvido em cada frente do projeto, bem como o endereço da plataforma em ambiente de produção.

A.1 Código-Fonte (Repositórios GitHub)

O projeto foi segmentado em três repositórios principais para garantir a modularidade e facilitar o gerenciamento das camadas de *firmware*, *backend* e *frontend*:

- **Firmware:** <<https://github.com/arlansantos/hive-management-firmware>>
- **Backend:** <<https://github.com/arlansantos/hive-management-backend>>
- **Frontend:** <<https://github.com/arlansantos/hive-management-frontend>>

A.2 Acesso à Aplicação

A plataforma encontra-se disponível para acesso público através do endereço:

- **Colmeia Digital:** <<https://www.colmeiadigital.site/>>

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Campina Grande - Código INEP: 25137409
	R. Tranquilino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
	CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Restrito

Versão Final do TCC

Assunto:	Versão Final do TCC
Assinado por:	Arlan Santos
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Restrito
Hipótese Legal:	Informação Pessoal (Art. 31 da Lei no 12.527/2011)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Arlan da Silva Santos, ALUNO (202011250027) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE, em 13/01/2026 15:24:43.

Este documento foi armazenado no SUAP em 13/01/2026. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1728707
Código de Autenticação: 8cb97976f6

