



**INSTITUTO  
FEDERAL**  
Brasília

Instituto Federal de Brasília

Campus Brasília

Curso Superior de Tecnologia em Sistemas para Internet

**APPIS: SENSORIAMENTO DE APIÁRIOS**

**Por**

***IGOR MICHELS GONÇALVES DE CARVALHO***

***JOÃO ALEXANDRE MARTINS BORGES***

**Tecnólogo**

BRASÍLIA

2025

Igor Michels Gonçalves de Carvalho  
João Alexandre Martins Borges

**APPIS: SENSORIAMENTO DE APIÁRIOS**

*Trabalho apresentado ao Curso Superior de Tecnologia em  
Sistemas para Internet da Instituto Federal de Brasília como  
requisito parcial para obtenção do grau de Tecnólogo em  
Sistemas de Internet .*

*Orientador: Dr. Caio Moura Daoud*

BRASÍLIA  
2025

C331a Carvalho, Igor Michels Gonçalves de.

APPIS: sensoriamento de apiários. / Igor Michels Gonçalves de Carvalho, João Alexandre Martins Borges ; orientação Caio Moura Daoud ; coorientação Tiago Henrique Faccio Segato. – Brasília, 2025.

63 f. : il. color. ; 30 cm.

Trabalho de Conclusão de Curso (Graduação) – Instituto Federal de Educação, Ciência e Tecnologia de Brasília, Tecnologia em Sistemas para Internet, 2025.

Orientador: Caio Moura Daoud ; Coorientador: Tiago Henrique Faccio Segato.

1. Apicultura. 2. Monitoramento remoto. 3. Arduino. 4. Internet das coisas. 5. Não invasiva. I. Borges, João Alexandre Martins. II. Daoud, Caio Moura, orient. III. Segato, Tiago Henrique Faccio, coorient. IV. Instituto Federal de Brasília. V. Título.

CDU 004.42:638.1

Igor Michels Gonçalves de Carvalho  
João Alexandre Martins Borges

**APPIS: Sensoriamento de Apiários**

Trabalho de conclusão de curso de graduação apresentado a Coordenação do Curso Superior de Tecnologia em Sistemas de Internet do Instituto Federal de Brasília – Campus Brasília, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas de Internet.

Aprovado em: \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_.

**BANCA EXAMINADORA**

---

Prof. Dr. Caio Moura Daoud  
Computação/IFB

---

Prof. Me. Tiago Henrique Faccio Segato  
Computação/IFB

---

Prof. Dr. Marco Rogério Calheira Lima  
Computação/IFB

---

Prof.<sup>a</sup> Dr.<sup>a</sup> Sylvana Karla da Silva de Lemos Santos  
Computação/IFB

BRASÍLIA  
2025

## Resumo

CARVALHO, Igor Michels Gonçalves de; BORGES, João Alexandre Martins. APPIS: Sensoriamento de apiários. 2025. 63 f. Trabalho de Conclusão de Curso (Graduação) – Tecnólogo em Sistemas para Internet. Instituto Federal de Brasília – Campus Brasília. Brasília/DF, 2025.

A apicultura, além de ser uma prática rentável, desempenha um papel fundamental na preservação ambiental, consistindo no cultivo de abelhas para a produção de mel, cera de abelha, pólen e geleia real. A manutenção adequada desses polinizadores é crucial para o equilíbrio da vida vegetal. No entanto, fatores adversos, como o aumento da temperatura, dos níveis de umidade e do peso da colmeia, têm impactado negativamente as populações de abelhas. Para enfrentar esses desafios, esta pesquisa propõe a confecção e implementação de um sistema de monitoramento remoto baseado em Internet das Coisas (IoT) utilizando a plataforma Arduino em apiários. Essa abordagem possibilitará uma avaliação não invasiva das atividades das abelhas, fornecendo suporte aos apicultores na gestão eficiente das colmeias. Dessa forma, busca-se minimizar os efeitos do calor, umidade, entre outros, nas colônias, contribuindo para a preservação desses polinizadores essenciais.

**Palavras-chave:** Apicultura. Monitoramento remoto. Arduino. Internet das Coisas. Não invasiva

## Abstract

CARVALHO, Igor Michels Gonçalves de; BORGES, João Alexandre Martins. APPIS: Sensoriamento de apiários. 2025. 63 f. Trabalho de Conclusão de Curso (Graduação) – Tecnólogo em Sistemas para Internet. Instituto Federal de Brasília – Campus Brasília. Brasília/DF, 2025.

Beekeeping, in addition to being a profitable practice, plays a key role in environmental preservation, consisting of the cultivation of bees for the production of honey, beeswax, pollen and royal jelly. Proper maintenance of these pollinators is crucial for the balance of plant life. However, adverse factors, such as increased temperature, humidity levels, and hive weight, have negatively impacted bee populations. To face these challenges, this research proposes the design and implementation of a remote monitoring system based on Internet of Things (IoT) using the Arduino platform in apiaries. This approach will enable a non-invasive assessment of bee activities, providing support to beekeepers in the efficient management of hives. In this way, the aim is to minimize the effects of heat, humidity, among other factors, on the colonies, contributing to the preservation of these essential pollinators.

**Keywords:** Beekeeping. Remote monitoring. Arduino. Internet of Things. Non-invasive.

## Lista de Figuras

3.1	Visão Protoboard do projeto . . . . .	21
3.2	Visita ao apicultor . . . . .	21
3.3	Colmeia presente na visita . . . . .	22
3.4	Visita ao apicultor . . . . .	22
4.1	ESP32 . . . . .	26
4.2	Arduino Uno R3 SMD . . . . .	28
4.3	Sensor DHT22 . . . . .	30
4.4	Célula de carga 50kg . . . . .	31
4.5	Amplificador Operacional . . . . .	32
4.6	Ponte de Wheatstone . . . . .	32
5.1	Esquema elétrico . . . . .	34
5.2	Código para firebase . . . . .	35
5.3	Código para firebase . . . . .	36
5.4	Código Sensor DHT22 . . . . .	39
5.5	Circuito DHT22 . . . . .	40
5.6	Teste na caixa de madeira . . . . .	41
5.7	Teste na caixa de plástico . . . . .	41
5.8	Teste na caixa de papelão . . . . .	42
5.9	Código para conexão . . . . .	43
5.10	Teste da placa ESP32 . . . . .	45
5.11	Placa de madeira base da balança . . . . .	46
5.12	Suporte 3D . . . . .	46
5.13	Visão da parte de baixo da balança . . . . .	47
5.14	Circuito teste de duas células de carga 50kg . . . . .	47
5.15	Inicialização do código . . . . .	48
5.16	Definição do fator de calibração . . . . .	48
5.17	Teste realizado com o celular . . . . .	49
5.18	Valores referente ao peso do celular . . . . .	49
5.19	Teste realizado com a tampa do apiário . . . . .	50
5.20	Valores referente ao peso da tampa do apiário . . . . .	50
5.21	Teste realizado com a base do apiário . . . . .	51
5.22	Valores referente ao peso da base do apiário . . . . .	51
5.23	Alteração no banco de dados do firebase . . . . .	52
5.24	Disposição idealizada dos sensores no apiário . . . . .	53
5.25	Variação lida do peso durante o teste . . . . .	54

5.26	Varição lida da temperatura durante o teste . . . . .	54
5.27	Varição lida da umidade durante o teste . . . . .	55
5.28	Sistema protegido com a caixinha de plástico . . . . .	55
5.29	Sistema montado em prateleira . . . . .	56
5.30	Gráfico da variação de temperatura . . . . .	57
5.31	Gráfico da variação de umidade . . . . .	58

## Lista de Tabelas

3.1	Cronograma . . . . .	20
3.2	Tabela de trabalhos correlatos. . . . .	23
3.3	Tabela com materiais dos trabalhos correlatos. . . . .	25

## Lista de Acrônimos

<b>IBGE</b>	Instituto Brasileiro de Geografia e Estatística . . . . .	12
<b>EMBRAPA</b>	Empresa Brasileira de Pesquisa Agropecuária . . . . .	13
<b>IoT</b>	Internet of Things . . . . .	14
<b>ESBMET</b>	Encontro Sul Brasileiro de Meteorologia . . . . .	18
<b>RAM</b>	Random Access Memory . . . . .	24
<b>HDMI</b>	High-Definition Multimedia Interface . . . . .	24
<b>USB</b>	Universal Serial Bus . . . . .	24
<b>SD</b>	Secure Digital Card . . . . .	24
<b>GSM</b>	Global System for Mobile Communications 2G . . . . .	24
<b>GPRS</b>	General Packet Radio Service . . . . .	24
<b>SMS</b>	Short Message Service . . . . .	24
<b>FTP</b>	File Transfer Protocol . . . . .	24
<b>UEPA</b>	Unidade de Extração de Produtos Apícolas . . . . .	24
<b>EAJ</b>	Escola Agrícola de Jundiá . . . . .	24
<b>UFRN</b>	Universidade Federal do Rio Grande do Norte . . . . .	24
<b>LARM</b>	Laboratório de Automação e Robótica Móvel da Universidade Federal de Santa Catarina . . . . .	24
<b>GPIO</b>	General-Purpose Input/Output . . . . .	27
<b>PWM</b>	Pulse Width Modulation . . . . .	27
<b>ROM</b>	Read-Only Memory . . . . .	27
<b>SPI</b>	Serial Peripheral Interface . . . . .	27
<b>BLE</b>	Bluetooth Low Energy . . . . .	27
<b>STA</b>	Station . . . . .	27
<b>AP</b>	Access Point . . . . .	27
<b>CPU</b>	Central Processing Unit . . . . .	27
<b>SMD</b>	Surface-Mount Device . . . . .	28
<b>EEPROM</b>	Electrically-Erasable Programmable Read-Only Memory . . . . .	28
<b>IDE</b>	Integrated Development Environment . . . . .	28
<b>A/D</b>	Analógico / Digital . . . . .	30

<b>mV</b>	milivolts . . . . .	31
<b>CAD</b>	Computer-Aided Design and Drafting . . . . .	33
<b>BaaS</b>	Backend as a Service . . . . .	34
<b>URL</b>	Uniform Resource Locator . . . . .	36
<b>MQTT</b>	Message Queuing Telemetry . . . . .	42
<b>CNC</b>	Computer Numeric Control . . . . .	45
<b>IFB</b>	Instituto Federal de Brasília . . . . .	45
<b>API</b>	Application Programming Interface . . . . .	52

## Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Tema . . . . .	13
1.2	Problema . . . . .	13
1.2.1	Objetivo geral . . . . .	14
1.2.2	Objetivos específicos . . . . .	14
1.3	Estrutura do TCC . . . . .	14
1.3.1	Classificação da Pesquisa . . . . .	15
<b>2</b>	<b>Conceitos gerais e revisão da literatura</b>	<b>16</b>
2.1	Internet das Coisas . . . . .	16
2.2	Sensores . . . . .	16
2.3	Importância da apicultura . . . . .	16
2.4	Termorregulação da colônia . . . . .	17
2.5	Peso . . . . .	17
<b>3</b>	<b>Metodologia</b>	<b>19</b>
3.1	Cronograma . . . . .	19
3.2	Descrição da proposta . . . . .	20
3.3	Pesquisa de campo . . . . .	21
3.4	Trabalhos correlatos . . . . .	23
<b>4</b>	<b>Materiais</b>	<b>26</b>
4.1	ESP32 . . . . .	26
4.2	Placa Arduino Uno R3 SMD . . . . .	28
4.3	Sensor DHT22 . . . . .	29
4.4	Célula de carga 50kg . . . . .	30
4.4.1	Módulo HX711 . . . . .	31
4.4.2	Ponte de Wheatstone . . . . .	32
<b>5</b>	<b>Experimentos Realizados</b>	<b>33</b>
5.1	Esquema elétrico do projeto . . . . .	33
5.2	Conexão com firebase . . . . .	34
5.3	Primeiros testes . . . . .	38
5.3.1	Teste com DHT22 . . . . .	39
5.3.2	Teste com ESP32 . . . . .	42
5.3.3	Teste com a célula de carga 50kg . . . . .	45
5.3.4	Testes com sensores e firebase . . . . .	52

5.4	Posicionamento dos sensores . . . . .	52
5.5	Teste contínuo . . . . .	54
5.6	Teste em campo . . . . .	55
<b>6</b>	<b>Conclusão</b>	<b>59</b>
	<b>Referências</b>	<b>60</b>

# 1

## Introdução

A apicultura consiste no cultivo de abelhas, com o objetivo de produção de mel, própolis, geleia real, pólen e cera de abelha (Incaper, 2016). É uma atividade essencialmente ecológica e rentável, que pode ser desenvolvida em praticamente qualquer espaço geográfico que possui condições de solo e clima favoráveis, e uma vegetação rica em floradas (Santos; Souza Ribeiro, 2009), como no caso da flora abundante brasileira.

As abelhas possuem um importante papel para a manutenção, principalmente, da vida vegetal, e infinitamente mais importante que apenas produzir mel. De acordo com Souza, Evangelista-Rodrigues e Caldas Pinto (2007) e com Possani et al. (2019), as abelhas são os principais e mais eficientes agentes polinizadores, responsáveis pela reprodução e perpetuação de espécies vegetais, incluindo os alimentos que nós consumimos, além de serem também um grande agente para conservar o meio ambiente e manter o equilíbrio dos ecossistemas. O pólen é de suma importância para o desenvolvimento da colmeia por se tratar da principal fonte proteica das abelhas, garantindo assim o desenvolvimento familiar das abelhas.

Além de sua relevância no ecossistema é importante ressaltar sua contribuição na economia do país. Conforme dados levantados pelo Instituto Brasileiro de Geografia e Estatística (IBGE), a produção de mel no Brasil atingiu 55,8 milhões de toneladas em 2022, resultando em um valor de produção de 854,4 milhões de reais. No total, 3991 (71,62%) municípios registraram produção de mel. De acordo com a Agrostat Brasil, durante o período de janeiro a julho de 2021, as agroindústrias brasileiras da apicultura exportaram 34.489 toneladas de mel in natura, registrando um aumento de 34,5% em relação ao mesmo período de 2020. O faturamento em dólares alcançou a marca de aproximadamente 118 milhões de dólares, representando um crescimento de 2,3 vezes em comparação com o valor obtido no mesmo período do ano anterior.

As abelhas seguem padrões estreitos de temperatura dentro do ninho. Elas possuem a capacidade de manter a temperatura das colônias entre 33°C a 36°C, contudo, a média é de 34,5°C (Jones; Oldroyd, 2006). Variações de temperatura podem comprometer o desenvolvimento das larvas e/ou pupas (Jones; Oldroyd, 2006).

Além da temperatura interna, fatores como a concentração da umidade da colméia afetam diretamente a colônia, podendo ocasionar óbitos de abelhas na fase adulta ou larval (Ellis et al., 2008). Segundo Human, Nicolson e Dietemann (2006), as abelhas possuem a capacida-

de de influenciar a umidade da colônia, contudo, com restrições nos mecanismos de regulação, já que faz-se presente variações de temperaturas em diversos locais do ninho.

A “Cartilha de Boas Práticas de Manejo Apícola” desenvolvida pela Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) ressalta a importância do mantimento de condições adequadas para a criação de abelhas, já que a escassez de recursos apibotânicos e/ou condições adversas como: temperatura, umidade do ar, falta de água entre outros, podem culminar em um comportamento enxameador por abandono, na qual as abelhas migram para outro local, garantindo dessa forma a sobrevivência da colônia, podendo dessa forma, segundo Almeida (2008), causar grande prejuízo aos apicultores.

### **1.1 Tema**

A temática neste trabalho aborda como captação de dados relevantes como temperatura, umidade e peso da colônia de abelhas auxiliam e facilitam o trabalho dos apicultores. Os dados coletados permitem que a abertura e revista presencial das colmeias para averiguar as condições do enxame e da produção de mel não seja necessária, evitando a culminação do caráter enxameador por abandono e garantindo melhores condições de vida das abelhas.

### **1.2 Problema**

Condições adversas de falta de água, temperatura, estresse, umidade relativa do ar, falta de alimento, entre outras, podem culminar no comportamento enxameador por abandono, o que vem a causar grandes prejuízos aos apicultores (Almeida, 2008).

Segundo Potts et al. (2010), fatores como destruição do habitat natural desses seres, desmatamento da biodiversidade, mudanças climáticas e utilização de agrotóxicos são os principais desafios que provocam a queda no número de abelhas em todo o mundo, comprometendo a polinização. Esses fatores podem agir conjunta ou separadamente, e podem exercer mais efeitos em certas situações e menos em outras, no entanto, o resultado é o desaparecimento da colônia (Da Rosa et al., 2017).

A fim de controlar esse comportamento da colônia, estudos de monitoramento e registro das atividades diárias, como, principalmente temperatura e umidade, podem auxiliar na identificação desses eventos (Almeida, 2008). Para conhecer a situação de uma colônia de abelhas é necessário uma verificação local e invasiva, o que acaba culminando em uma mudança de comportamento desses insetos.

Portanto, a monitoração por sensores e, conseqüentemente, o registro dos dados disponíveis ao apicultor tem se mostrado uma atividade eficiente (Silva, 2017). Alguns trabalhos e suas descobertas foram e serão utilizados como parâmetro para se buscar melhorias nesse campo. Trabalhos como o de Murphy et al. (2015) desenvolveram um projeto de captação de imagens e som das câmaras internas da colmeia, bem como um sistema de alerta para eventos

de emergência detectados por meio de envio de SMS. Já o trabalho de Abreu e Perez (2017) captava amostras de registro de temperatura e umidade para mapear os períodos de alterações climáticas, definindo um intervalo de tempo em que o sensor poderia entrar em *sleep mode* (modo de suspensão) para economizar o consumo de energia.

Com base no contexto apresentado, este trabalho propõe o desenvolvimento e implementação de uma tecnologia de monitoramento remoto e não invasivo para apiários, que tem como consequência impulsionar a produção de mel e a atividade de polinização, com benefícios tanto para a economia quanto para o meio ambiente.

### 1.2.1 Objetivo geral

Desenvolver um sistema de monitoramento e registro de variáveis ambientais para uma colmeia apícola, utilizando *Internet of Things (IoT)*. O objetivo é aprimorar a gestão, fornecendo mais controle da colmeia ao apicultor e reduzir a necessidade de processos invasivos, como abertura das caixas. Serão utilizados sensores de temperatura, umidade e peso, conectados a uma placa responsável por enviar as informações coletadas para um banco de dados.

### 1.2.2 Objetivos específicos

Para atingir o objetivo geral, foram estabelecidos alguns objetivos específicos a serem alcançados e que nos orientam nas etapas do nosso desenvolvimento:

- Realizar pesquisa bibliográfica para compreender os fundamentos do sensoriamento: Fazer uma revisão abrangente da literatura relacionada à apicultura, monitoramento de colmeias e tecnologias IoT para embasar a implementação do sistema;
- Integrar os sensores junto a placa microcontroladora em processo controlado, para o monitoramento preciso do ambiente em tempo real;
- Elaborar os códigos referentes aos sensores;
- Programar a placa central para envio dos dados ao servidor nuvem;
- Conduzir experimentos práticos e validação da eficácia em condições reais de apicultura após a montagem do circuito.

## 1.3 Estrutura do TCC

O presente trabalho está dividido em 5 tópicos, cada um com suas especificidades e organizados de acordo com as coerências teóricas do estudo e compreensão das objetividades. Na primeira parte estão apresentadas a introdução do estudo, o tema, a problemática, os objetivos (gerais e específicos), a estrutura do trabalho, juntamente a classificação da pesquisa. Na segunda parte do estudo, encontram-se os conceitos gerais e revisão de literatura acerca dos trabalhos

e projetos já desenvolvidos nessa área. Na terceira parte, é descrito a metodologia, a pesquisa de campo realizada, assim como os materiais que virão a ser utilizados no desenvolvimento do projeto. Na quarta parte do trabalho, está presente o campo referente aos resultados obtidos, com a análise dos mesmos e sua apresentação. Na quinta e última parte do trabalho serão apresentadas as conclusões do estudo e apresentação dos possíveis trabalhos a serem realizados futuramente.

### 1.3.1 Classificação da Pesquisa

A seguir será apresentado como essa pesquisa será classificada:

- Quanto à natureza da pesquisa, o trabalho será de natureza aplicada;
- Quanto à abordagem: Será de caráter quantitativo, pois será coletado dados (conceitos que podem ser medidos);
- Quanto aos objetivos, considera-se o trabalho como exploratório;
- Quanto aos métodos, o trabalho consiste em pesquisa experimental, juntamente com *survey* (levantamento), estudo de caso e pesquisa bibliográfica possibilitando o entendimento do que já foi produzido.

## 2

### **Conceitos gerais e revisão da literatura**

#### **2.1 Internet das Coisas**

A Internet das Coisas (ou em inglês, *Internet of Things*, IoT) é um conceito que afirma que a internet irá englobar objetos do nosso dia a dia, permitindo que possam ser acessados e controlados via internet (Mattern; Floerkemeier, 2010). A Internet das Coisas é usada em diversas áreas, tanto em ambientes acadêmicos, comerciais e domésticos. No contexto do monitoramento de colmeias, a aplicação de IoT oferece uma abordagem inovadora para acompanhar e compreender as atividades das abelhas de maneira remota e eficiente.

#### **2.2 Sensores**

Segundo o site Nexxto (2020), os dispositivos de sensoriamento são como os "sentidos" de um sistema de controle e movimento, interagindo com o ambiente, captando informações de algum tipo de energia (luz, temperatura, movimento etc), podendo medir essa energia variável e transformá-la em sinais, posteriormente sendo convertidos em dados. Desempenham um papel crucial no ecossistema de monitoramento, sendo responsáveis pela captação de dados ambientais e comportamentais das colmeias. Sensores de temperatura, umidade e peso são exemplos pertinentes, proporcionando informações precisas que contribuem para a análise abrangente do ambiente das abelhas.

#### **2.3 Importância da apicultura**

A apicultura prevalece como uma atividade que apresenta contribuições significativas com as dimensões do desenvolvimento sustentável, sejam elas econômicas, porque gera renda e lucratividade superior comparada a outras atividades rurais; sociais, porque pode ser um fator determinante para a redução do êxodo rural; e ambiental, porque não degrada os recursos naturais e contribui com o aumento da conscientização acerca da preservação da natureza (Lourenço; Oliveira Cabral 2016).

Ainda de acordo com Lourenço e Oliveira Cabral (2016), é uma atividade de baixos gastos de manutenção e que não demanda muito tempo, bem como de fácil execução, não neces-

sitando de práticas sofisticadas.

Dentre os principais benefícios ao meio ambiente proporcionados pela atividade apícola podem ser destacados: é uma atividade de pouco impacto ambiental; a polinização das abelhas maximiza a produtividade de atividades vegetais; contribui para a preservação e perpetuação da biodiversidade, aumentando a variabilidade genética das plantas polinizadas; além de colaborar na recuperação de áreas degradadas e na regeneração natural (Araújo et al., 2015).

Economicamente, o ramo apícola é visto com grande potencial, responsável por oferecer produtos diversificados como mel, cera, própolis, geleia real, pólen, composição de produtos farmacêuticos e cosméticos, apresentando uma grande procura tanto do mercado interno quanto do mercado externo (Tomazini; Fátima Grossi, 2019).

## 2.4 Termorregulação da colônia

A abelha consiste num animal pecilotérmico, isso significa que sua temperatura corporal apresenta-se próxima à temperatura ambiente, sofrendo algumas variações devido a realização de suas atividades (Carvalho, 2009). Temperaturas baixas normalmente dificultam suas atividades, em contrapartida, temperaturas mais altas estimulam o animal (Almeida, 2008).

De acordo com Budel (1968) e Heran (1968), as abelhas do gênero *Apis* conservam calor por aglomeração de indivíduos, além de utilizarem da contração muscular para conservar calor e controlar a temperatura através da ventilação. Características estas e dentre outras que possibilitaram a expansão desse gênero para regiões temperadas.

As abelhas conseguem termorregular e manter, na grande maioria das vezes, a temperatura interna em torno de 35 graus Celsius e uma umidade relativa de 80% (Almeida, 2008). Como diz Almeida (2008) em seu estudo, elas tentam termorregular até uma certa temperatura (entre 41 e 43 graus Celsius), sendo que após este limite elas em geral abandonam a colônia deixando cria e alimento, mesmo que essas duas variáveis estejam em abundância.

A umidade relativa é dependente da temperatura, podendo ser passivamente (através da transpiração dos indivíduos da colmeia e capacidade do efeito do néctar) ou ativamente regulada (coleta de água e transpiração) de acordo com o trabalho de Almeida (2008). Já na pesquisa de Human, Nicolson e Dietemann (2006) encontrou que as abelhas operárias podem ajustar a umidade a limites sub-ótimos, constando ainda que em altas temperaturas, a coleta de água auxilia no resfriamento evaporativo, contribuindo conseqüentemente para a homeostase da colônia.

## 2.5 Peso

O peso de uma colônia é um reflexo direto de seu estado de saúde e produtividade (Fitzgerald et al., 2015). Segundo Fitzgerald et al. (2015), fatores como temperatura ambiente, quantidade de horas de contato com o sol, variação de temperatura e umidade podem

afetar o peso de uma colmeia.

Segundo os experimentos realizados por Vieira et al. (2019) e apresentados no VIII Encontro Sul Brasileiro de Meteorologia (ESBMET), durante o período monitorado houve variação no peso das colmeias com ocorrência de alternâncias negativas e positivas, apesar da disponibilidade de néctar durante as floradas. Nos dias de variações negativas do peso, a temperatura do ar estava mais elevada com máximas acima de 30 graus Celsius, alta umidade relativa do ar, baixa pressão atmosférica e chuvas esparsas, inferindo no comportamento das colmeias e das abelhas.

O objetivo de medir o peso de uma colônia é determinar quando os estoques de mel estarão com sua capacidade máxima durante os períodos de produção do material, o que consiste numa atividade muito difícil de saber com os métodos tradicionais e imprecisos (Fitzgerald et al., 2015), além de darem indícios sobre os níveis de estresse (Vieira et al., 2019).

## 3

### Metodologia

A metodologia que será adotada no desenvolvimento da tecnologia de monitoramento de apiários é composta por várias etapas que combinam pesquisa sobre o assunto, desenvolvimento tecnológico, experimentação prática e interação com usuários finais. Consiste numa pesquisa de caráter experimental aplicada, visto que haverá a manipulação controlada (*in lock*) das variáveis durante o desenvolvimento do sistema e da coleta de dados.

Será adotada uma abordagem iterativa, combinando com os princípios do modelo incremental. O modelo incremental permite uma maior flexibilidade para adaptar o projeto conforme o surgimento de novas necessidades e com novas informações sendo obtidas durante a pesquisa e experimentação. A abordagem iterativa permite justamente que cada incremento seja desenvolvido em ciclos de iteração, permitindo que o projeto possa ser dividido em pequenas etapas que serão repetidas e refinadas ao longo do tempo. Uma evolução contínua da solução e uma entrega do produto final robusta e eficiente.

Para a elaboração do presente projeto foi necessário fazer uma pesquisa fundamentada para embasar cada etapa do desenvolvimento e para fornecer o conhecimento adequado sobre o contexto da apicultura. Desse modo foram usadas fontes como teses, artigos científicos e monografias, bem como coletas de informações do tema com pesquisas de campo, como será descrito com detalhes na seção 3.3 desse trabalho. Os livros e artigos que foram usados são encontrados em repositórios acadêmicos e base de dados científicas, além de bibliotecas universitárias também.

Os resultados deste trabalho, que são de natureza quantitativa, proporcionam a visualização detalhada e mensurável do desempenho do sistema de monitoramento de colmeias. As informações de uma caixa de abelhas como, temperatura, umidade e peso podem ditar as atividades a serem realizadas e cuidados a serem tomados.

#### 3.1 Cronograma

O cronograma apresenta uma sequência de atividades ao longo do tempo, com cada mês do desenvolvimento, que nos guiará para o cumprimento das atividades, que estão na primeira coluna à esquerda do cronograma.

Ano	2023/2024										
Mês	ago	set	out	nov	dez	jan	fev	mar	abr	mai	jun
Revisão bibliográfica	X	X	X	X	X	X					
Definição de metodologia		X									
Aquisição de materiais			X	X	X						
Montagem de hardware				X	X	X					
Desenvolvimento de código					X	X	X				
Teste e Experimentos							X	X			
Coleta e análise de dados							X	X	X		
Escrita da documentação								X	X	X	
Apresentação a banca											X
Revisão final e correções											X

Tabela 3.1 Cronograma

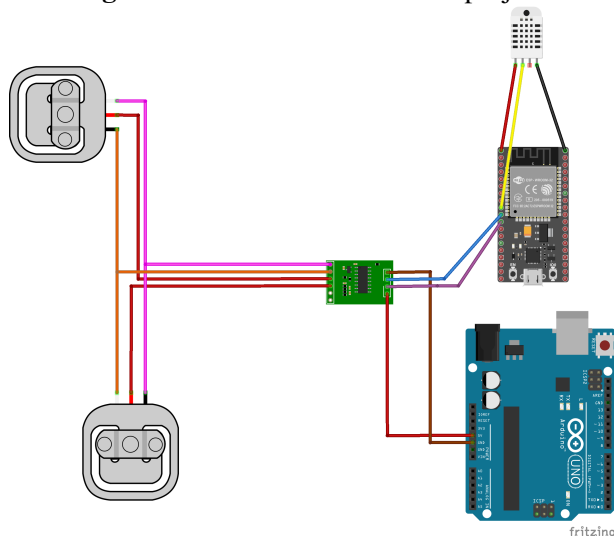
Inicia-se com a revisão bibliográfica, seguida pela definição da metodologia e aquisição de materiais. Posteriormente, concentra-se na montagem do hardware, desenvolvimento do código e realiza testes e experimentos. As etapas finais incluem coleta e análise de dados, escrita da documentação e apresentação à banca, finalizando com a revisão final e correções necessárias. Este planejamento garante uma execução eficiente e progressiva do projeto.

Há de se dizer que durante a execução do projeto houve um período de greve que impactou diretamente as atividades anteriormente descritas no cronograma. Algumas atividades foram prolongadas durante esse tempo e outras foram adiadas para os meses seguintes. Apesar da situação, o projeto foi concluído sem comprometer a qualidade ou objetivos propostos.

### 3.2 Descrição da proposta

O propósito do projeto é a concepção e implementação de um protótipo funcional, que se revela dinâmico e propenso a aprimoramentos contínuos. A visão orientadora consiste em desenvolver um circuito integrado personalizado para cada caixa do apiário, refletindo a modularidade e adaptabilidade inerentes ao sistema.

A infraestrutura será estruturada em locais estrategicamente posicionados para facilitar o acesso à rede Wi-Fi. Essa escolha não apenas simplifica a comunicação bidirecional eficiente com o banco de dados não-relacional, mas também viabiliza uma interação intuitiva entre o sistema e a interface projetada para os apicultores. Essa interface, em desenvolvimento por colaboradores parceiros, visa proporcionar uma experiência amigável e informativa aos usuários finais.

**Figura 3.1** Visão Protoboard do projeto

Fonte: Autor

### 3.3 Pesquisa de campo

A fim de ter uma melhor compreensão da rotina de um apiário, foi realizada uma pesquisa de campo, podendo realizar as observações necessárias com proximidade, do espaço, das dimensões, e do funcionamento necessário para a criação de abelhas.

**Figura 3.2** Visita ao apicultor

Fonte: Autor

Primeiramente foram visitados dois lugares diferentes, uma chácara na região da Fercal no quilômetro 12 da DF-150 que tinha 3 colmeias, e depois uma casa numa região próxima, na qual possuía em seus fundos 2 colmeias. O apicultor entrevistado relatou à equipe que está no ramo da apicultura por influência familiar, executando como hobby. Ele possui à sua disposição: 2 trajes de apicultor, um fumigador e uma centrífuga. O entrevistado também relatou que iria fazer uso de alimentadores que ainda seriam instalados.

Quanto à estrutura dos apiários, as caixas são de madeira pinos e estão suspensas em um cavalete de aço, com potes com água nos pés para evitar que as formigas subam.

**Figura 3.3** Colmeia presente na visita

Fonte: Autor

Para averiguar o estado de saúde de suas colmeias, devido a falta de coleta de dados de forma remota, o apicultor é obrigado a realizar ações intrusivas, as quais podem culminar em processos enxameatórios ao longo do tempo. Além disso, para eventual controle de doenças, fomos notificados pelo apicultor sobre a necessidade de limpeza das caixas Langstroth.

Com a apresentação da proposta do nosso equipamento, o apicultor validou a importância e demanda do produto além de se mostrar interessado em colaborar com a parte prática e implementação de protótipos em suas colmeias. Atualmente os únicos dados que eles anotam e guardam são as datas que trouxeram as colmeias. Sensores que captam dados de umidade, temperatura e peso das abelhas mostraram ser relevantes e essenciais para o apicultor.

**Figura 3.4** Visita ao apicultor

Fonte: Autor

Adicionalmente, foi realizada uma visita em 19 de dezembro de 2024, ao apiário do Instituto Federal de Brasília (IFB) no campus Planaltina, onde houve a oportunidade de conversar

com um professor que nos apresentou a área destinada à criação de abelhas da espécie *Apis mellifera*. Durante a visita, foi observado que a região, localizada longe dos blocos principais do campus, não possui uma conexão estável com a internet, tornando necessária a utilização de uma antena para estabelecer comunicação eficiente. O professor validou a proposta do projeto por meio de um entrevista e conversas detalhadas, demonstrando entusiasmo com a aplicação prática, e fez perguntas pertinentes sobre a funcionalidade e os detalhes das instalações. Além disso, ele contribuiu com sugestões valiosas e ofereceu uma caixa do modelo Langstroth, permitindo que fossem tiradas medidas e realizados testes em laboratório.

Posteriormente, foi realizada uma visita ao apicultor Bruno, que possui caixas de abelhas da espécie *Tetragonisca angustula* (Jataí). No local, o sistema foi montado e submetido a testes, permitindo observar sua funcionalidade em um ambiente real, cujos resultados estão na seção 5. Durante a interação, o apicultor demonstrou interesse nos dados coletados pelo sistema, apontando o ponto de contribuição do projeto para o monitoramento de condições ambientais e de saúde das abelhas em seu apiário, destacando a relevância de soluções tecnológicas que auxiliem na apicultura.

### 3.4 Trabalhos correlatos

Nos últimos anos, tem sido testemunhado um notável avanço no campo dos sistemas de monitoramento de apiários, revelando uma crescente preocupação e interesse pela saúde e produtividade das colônias de abelhas. Os trabalhos utilizados como referência neste trabalho foram escolhidos pelas suas notoriedades e citações em outros trabalhos pesquisados.

ID único	Título do trabalho	Ano
T1	Energy neutral platform for remote monitoring of beehive imagery and sound.	2015
T2	Beehiveior - sistema de monitoramento e controle de colmeias de produção apícola.	2016
T3	Redução do Consumo Energético de um Sensor Sem Fio Utilizado na Automação de um Abrigo de Cultivo.	2017
T4	Sistema de monitoreo de un apiario basado en el internet de las cosas.	2022

**Tabela 3.2** Tabela de trabalhos correlatos.

O trabalho de Murphy et al. (2015), com número de identificação T1 (como mostrado na Tabela 2.1) apresenta uma proposta de desenvolvimento de um sistema responsável por captar variáveis de sons, imagens e movimento, a fim de visualizar as ações da colônia principalmente na época do inverno. O projeto possui um controlador Raspberry Pi Model B com 512 MB de Random Access Memory (RAM), conexão High-Definition Multimedia Interface (HDMI), 2 portas Universal Serial Bus (USB); e para o armazenamento dos dados, um cartão Secure Digital Card (SD). Para conexão, os autores optaram pelo módulo SIM900 (SIMCom) *Global System for Mobile Communications 2G (GSM)/ General Packet Radio Service (GPRS)*, provendo comunicações por linha de telefone, Short Message Service (SMS) e operações de downloads e uploads *File Transfer Protocol (FTP)*. Os arquivos das imagens são salvos no cartão SD, podendo ser recuperados do cartão através da rede pelo módulo GSM/ GPRS e carregados para um servidor via FTP. Quanto a captação dos sons, uma vez que uma perturbação ocorre e se sustenta por alguns segundos, um alerta de texto é enviado ao apicultor informando que o apiário necessita de uma checagem. Já quanto a variância de movimento, alertas são enviados ao apicultor em caso de mudanças significativas.

O projeto desenvolvido por Dutra (2016) e identificado por T2, tem por objetivo ser capaz de auxiliar e dar maior embasamento aos estudos e pesquisas na área apícola, inicialmente atuando na Unidade de Extração de Produtos Apícolas (UEPA) da Escola Agrícola de Jundiá (EAJ) da Universidade Federal do Rio Grande do Norte (UFRN), além de facilitar as atividades cotidianas presentes no manejo das colmeias. O protótipo capta 5 variáveis da colmeia: temperatura, umidade, ruído, presença e contagem dos insetos. A placa utilizada é o Arduino UNO Rev 3 acoplado a um shield para fornecer conexão Ethernet. Apresenta uma interface Web para interação com o usuário, em que o mesmo é capaz de escolher com qual apiário registrado deseja trabalhar. Segundo o autor, a vantagem do projeto em relação aos demais produtos existentes é a não necessidade de hardware proprietário para realização da coleta de dados, sendo que qualquer pesquisador / apicultor poderá montar sua própria estrutura de coleta.

No trabalho com ID T3, de Abreu e Perez (2017), o protótipo foi desenvolvido no Laboratório de Automação e Robótica Móvel da Universidade Federal de Santa Catarina (LARM) com o objetivo de reduzir o consumo do sensor sem fio que é utilizado em experimentos de automação em um abrigo de cultivo em escala reduzida para realização de testes. O sistema é responsável pela coleta de variáveis como temperatura, umidade e luminosidade do ambiente, fazendo o envio desses dados para um controlador central através de um enlace de comunicação sem fio. O controlador central é uma placa Raspberry Pi Model 1, de processador Broadcom BCM2835, junto a ele acoplado um transceptor NRF24L01+ para recepção dos dados enviados pelo sensor e um driver para controle dos atuadores do projeto. O controlador central ainda é responsável por enviar os dados a uma plataforma IoT (Internet das Coisas) para armazenamento dos dados na nuvem. Foi instalado ainda o sistema operacional Raspbian que executa um sistema de monitoramento escrito em Python com framework Django, sendo responsável por controlar todo o sistema desenvolvido, fazer interação com o driver de controle dos atuadores e

fornecer uma interface web para o usuário interagir.

Desenvolvido por Echavarría et al. (2022), com identificação nomeada por T4, o protótipo foi projetado para ajudar e alertar os apicultores sobre as condições internas das colmeias, visto que o mercado apicultor da Colômbia está crescendo significativamente, especialmente em zonas rurais, possibilitando um acesso mais rápido e fácil ao estado da colmeia. O trabalho capta variáveis de ambiente como: temperatura, umidade, peso e ruído; e utiliza um microcontrolador ESP32 para comunicação com o servidor nuvem IoT (servidor ThingSpeak), responsável pelo armazenamento dos dados obtidos. Segundo o autor, esta seção da comunicação do microcontrolador e a plataforma IoT, é composta por um telefone celular que permite uma conexão direta com a rede sem fio, dando maior estabilidade ao sistema pois permite que ele seja portátil e tenha maior cobertura.

ID	Placa	Sensor(es)	Rede	Energia	Variáveis	Armazenamento
T1	Raspberry Pi Model B	Pi NoIR camera; SparkFun Electret Microphone Breakout; LIS331DLH	GSM/GPRS	Painel Solar	Som, Imagem, movimento	Cartão SD; Servidor FTP
T2	Arduino UNO Rev 3	DHT11, outros (modelos não informados)	Ethernet	Fonte de Tensão Contínua	Temperatura, umidade, ruído, presença e contagem de abelhas (infravermelho)	Cartão SD
T3	Raspberry Pi Model 1	DHT11; LDR	WAN	Fonte de Tensão Contínua	Temperatura, umidade, luminosidade	Servidor nuvem
T4	ESP32	DHT22; MAX9814; Célula de carga 20kg	Rede celular	Painel Solar; Bateria	Temperatura, umidade, peso e ruído	Servidor nuvem

**Tabela 3.3** Tabela com materiais dos trabalhos correlatos.

## 4

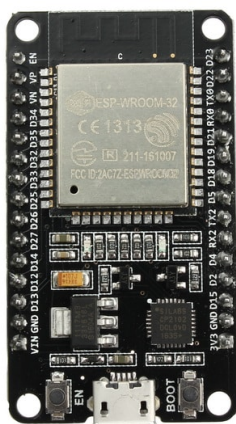
### Materiais

Além de coletar informações em tempo real sobre as colmeias, o presente projeto também tem como objetivo fazer o armazenamento, a análise e a amostragem desses dados, de forma que o apicultor consiga visualizar essas informações. As entregas são um sistema de monitoramento remoto inteligente, criação de relatórios quanto aos resultados coletados, divulgação do conhecimento e experiências. Neste capítulo, serão apresentados e descritos todos os componentes eletrônicos utilizados na confecção do protótipo.

#### 4.1 ESP32

O módulo ESP32, como mostrado na figura 4.1, apresenta-se como um componente principal para projetos que exigem conectividade e processamento integrado, abrangendo desde aplicações de automação residencial até dispositivos IoT industriais. Desenvolvido pela Espressif Systems, é uma evolução do ESP8266, trazendo melhorias significativas em termos de desempenho e capacidades.

**Figura 4.1** ESP32



Fonte: MakerHero

A placa é equipada com 32 bits que operam a uma velocidade de até 240 MHz. Isso proporciona uma capacidade de processamento considerável, permitindo que o módulo execute

tarefas simultaneamente sem comprometer o desempenho (Makerhero, 2022). O microprocessador Xtensa® Dual-Core 32-bit LX6 garante um desempenho computacional robusto necessário para executar algoritmos complexos de processamento de dados e controle em tempo real.

A funcionalidade do ESP32 se destaca pela integração nativa de conectividade Wi-Fi e Bluetooth, permitindo comunicação simultânea e eficiente com redes sem fio e dispositivos móveis. Este recurso é particularmente importante em sistemas que atualizam dados em tempo real e interação direta com smartphones ou outros módulos Bluetooth. O ESP32 possibilita a leitura e transmissão de dados de sensores para a nuvem, além da execução de ações remotas através de atuadores.

A Placa WiFi ESP32 Bluetooth é muito versátil quanto à sua programação, podendo ser programado diretamente pela IDE Arduino em C/C++, suporta MicroPython e CircuitPython, LUA e outras linguagens de programação de alto nível (Makerhero, 2022).

Especificações técnicas da placa, de acordo com o site da ELETROGATE (2020):

- Quantidade de pinos: 30;
- Central Processing Unit (CPU): Xtensa® Dual-Core 32-bit LX6;
- Read-Only Memory (ROM): 448 KBytes;
- RAM: 520 Kbytes;
- Flash: 4 MB;
- Modos de operação: Station (STA)/Access Point (AP)/STA+AP;
- Bluetooth: Bluetooth Low Energy (BLE) 4.2;
- Portas General-Purpose Input/Output (GPIO): 25;
- GPIO com funções de Pulse Width Modulation (PWM), I2C, Serial Peripheral Interface (SPI), etc;
- Tensão de operação: 4,5 - 9V;
- Taxa de transferência: 110 - 460800bps;
- Possui: Wireless padrão 802.11 b/g/n, Conexão Wifi 2.4Ghz (máximo de 150 Mbps), Conversor analógico digital (ADC);
- Clock máximo: 240MHz.

Visando o funcionamento do projeto, o ESP32 destaca-se pela sua capacidade de conectividade Wi-Fi e Bluetooth, permitindo comunicação em tempo real e controle remoto, bem

como sua integração com os componentes eletrônicos do protótipo, o que é essencial para aplicações de monitoramento contínuo. Sua robustez, documentação abrangente e o suporte ativo da comunidade global oferecem uma base sólida para o desenvolvimento de um sistema de monitoramento eficiente e confiável em projetos de IoT. Suas especificidades o tornam uma escolha excepcional para a implementação de soluções sofisticadas em diversos ambientes.

#### 4.2 Placa Arduino Uno R3 SMD

A placa microcontroladora Arduino Uno R3 SMD, como mostrado na figura 4.2, desempenha um papel central no âmbito do projeto de monitoramento inteligente de colmeias. Desenvolvida pela renomada Arduino LLC, essa placa proporciona uma plataforma flexível e economicamente acessível, primordial para a concepção de dispositivos eletrônicos interativos.

**Figura 4.2** Arduino Uno R3 SMD



Fonte: HU Infinito Componentes Eletrônicos

O Arduino Uno R3 SMD é dotado de um microcontrolador ATmega328P em formato Surface-Mount Device (SMD), atuando como o núcleo do sistema. Esta característica permite uma construção mais compacta e robusta. Sua função consiste em executar o código elaborado, desempenhando tarefas específicas como a leitura de dados provenientes dos sensores e o subsequente envio dessas informações para a nuvem.

O funcionamento do Arduino UNO R3 SMD ocorre com a gravação de códigos de programação na Electrically-Erasable Programmable Read-Only Memory (EEPROM), usando a linguagem de programação baseada em Wiring, e o software Arduino Integrated Development Environment (IDE), que permite a escrita, compilação e upload de códigos para a placa (ARDUINO, 2011). Os pinos de entrada e saída digital e analógica possibilitam a interação com uma gama variada de sensores, atuadores e outros dispositivos eletrônicos.

A placa é equipada com, além do microcontrolador citado acima:

- Tensão de operação de 5V;
- Tensão de entrada recomendada: 7 - 12V;
- Pinos digitais: 14 (dos quais 6 podem ser usados como saídas PWM);
- Pinos analógicos: 6;
- Corrente DC por pino de E/S: 20mA (40mA máxima);
- Memória Flash: 32KB (dos quais 0,5KB são usados pelo bootloader);
- SRAM: 2KB;
- EEPROM: 1KB;
- Frequência de clock: 16 MHz.

No contexto do projeto, a escolha do Arduino Uno R3 SMD revela-se crucial para a coleta de dados provenientes do sensor de temperatura DHT22 e da célula de carga utilizada, bem como para a integração eficiente com outros elementos do sistema. Sua confiabilidade comprovada, extensa documentação, custo acessível e a presença ativa de uma comunidade de suporte consolidada, fortifica a escolha do Arduino Uno como instrumento para o desenvolvimento de um sistema para monitoramento de apiários.

### 4.3 Sensor DHT22

O sensor de temperatura DHT22, também conhecido como AM2302, desempenha um papel essencial nesse projeto de monitoramento inteligente de colmeias. Este dispositivo compacto e acessível é capaz de medir tanto a temperatura quanto a umidade relativa do ambiente, oferecendo informações essenciais para entender o ecossistema do apiário.

O DHT22 opera por meio de um elemento de polímero capacitivo responsável por medir a umidade, e um termistor para medir a temperatura. Alimentado por uma corrente elétrica, o sensor aquece o termistor para medir a temperatura ambiente, ao mesmo tempo em que o elemento capacitivo absorve ou libera moléculas de água do ar circundante, alterando sua capacitância de acordo com a umidade (Makerhero, 2023).

Quando solicitado, o sensor realiza a leitura desses parâmetros e os converte em sinais elétricos. Em seguida, um microcontrolador, como o Arduino Uno, interpreta esses sinais e os traduz em valores de temperatura e umidade, prontos para serem utilizados no monitoramento.

**Figura 4.3** Sensor DHT22

Fonte: Eletrogate

O sensor apresenta as seguintes especificações:

- Tensão de operação de 3-5V (5,5V máximo);
- Faixa de medição de umidade: 0-100% de umidade relativa, com uma precisão de  $\pm 2-5\%$  UR;
- Faixa de medição de temperatura:  $-40$  a  $+80$  °C, com uma precisão de  $\pm 0.5$  °C;
- Intervalo mínimo entre leituras: 2 segundos;
- Corrente: Máximo de 2,5 mA durante o uso. 100mA a 150mA em modo de espera;
- 4 pinos: VCC (alimentação), DADOS (envio dos dados), N.C (não utilizado para esse projeto) e GND (conectado à terra).

A incorporação do DHT22 permite uma coleta precisa e em tempo real das condições térmicas e de umidade dentro das colmeias. Esses dados são fundamentais para compreender o comportamento das abelhas, garantindo um ambiente propício ao seu desenvolvimento. Além disso, a simplicidade de integração e a baixa demanda de recursos tornam o DHT22 uma escolha eficaz para projetos de monitoramento e sistemas de automação.

#### 4.4 Célula de carga 50kg

A Célula de Carga, conhecida também como *strain gage* ou *load cell*, é um componente crucial nesse sistema de monitoramento de colmeias. Trata-se de um transdutor responsável por variar a resistência de forma proporcional à deformação gerada pelo peso imposto sobre ela. Para trabalhar com esta célula é necessário um conversor Analógico / Digital (A/D) amplificador, sendo normalmente utilizado o HX711 (Circuito, 2022).

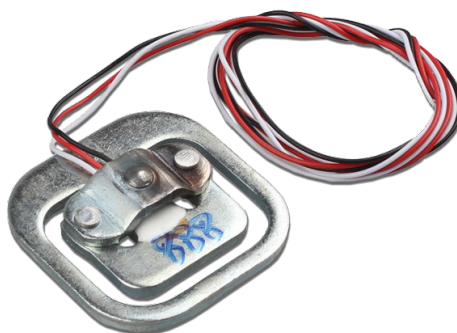
Em relação a variação de resistência, as células de carga têm um funcionamento parecido com potenciômetros. Elas possuem extensômetros que se deformam de acordo com a carga aplicada sobre sua área de contato, e essa deformação que permite a variação da resistência.

Neste projeto, a célula de carga é posicionada abaixo do apiário, medindo sua força peso. Conforme o peso da colmeia varia, a célula de carga responde gerando um sinal elétrico proporcional à carga.

O impacto deste sensor no projeto é significativo, pois a informação do peso é valiosa para avaliar a saúde da colônia, identificar mudanças na produção de mel e detectar potenciais problemas, como a diminuição de abelhas ou a necessidade de alimentação.

A integração da célula de carga no sistema, juntamente com outros sensores, proporciona uma visão abrangente das condições das colmeias, contribuindo para decisões mais informadas por parte dos apicultores.

**Figura 4.4** Célula de carga 50kg

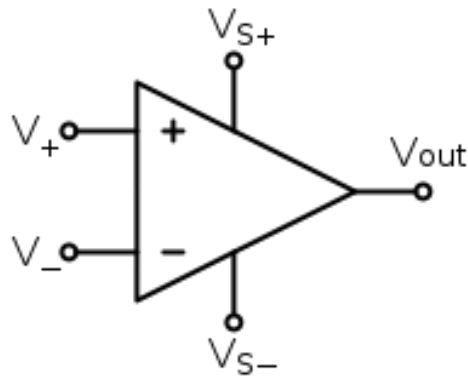


Fonte: Robocore

#### 4.4.1 Módulo HX711

O módulo HX711 é um conversor analógico-digital e amplificador operacional de 24 bits. Usado para amplificar o sinal da célula de carga de 50kg, além de converter o sinal analógico para um sinal digital. Essa prática torna-se necessária pois o sinal de saída da célula está na escala de milivolts (mV), portanto para que seja lido pelo microcontrolador, o sinal precisa ser amplificado para o nível adequado da placa.

Um amplificador operacional geralmente possui uma entrada variável e uma entrada constante, retornando, como saída, a diferença das tensões, de acordo com o ganho do amplificador:  $V_{out} = A * [ (V+) - (V-) ]$ . Em que “A” costuma ser fornecido como um valor fixo pelo fabricante para o desenvolvimento do circuito.

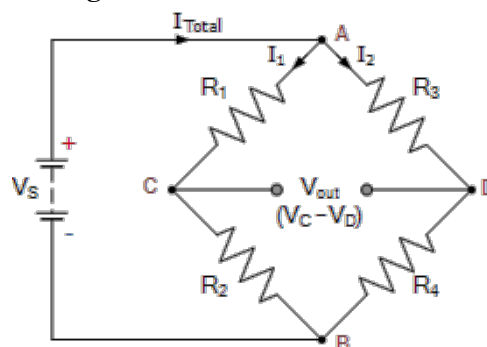
**Figura 4.5** Amplificador Operacional

Fonte: Wikipedia

#### 4.4.2 Ponte de Wheatstone

Os *strain gauges* são conectados em uma configuração de ponte de Wheatstone, que permite a conversão da alteração de resistência em um sinal elétrico de tensão (mV). Por meio deste recurso eletrodinâmico é possível identificar valores de resistência ôhmica desconhecida. Utilizada para determinar resistências muito pequenas, podendo ser utilizada para medir uma escala de mili-ohms.

O circuito da ponte nada mais é do que duas disposições série-paralelo de dois resistores, preferencialmente de mesmo valor. Com o valor igualado em todos os resistores da ponte, a diferença de potencial no centro da ponte quando medido entre as disposições de resistores deve ser igual a zero.

**Figura 4.6** Ponte de Wheatstone

Fonte: Eletronics Tutorials

## 5

### Experimentos Realizados

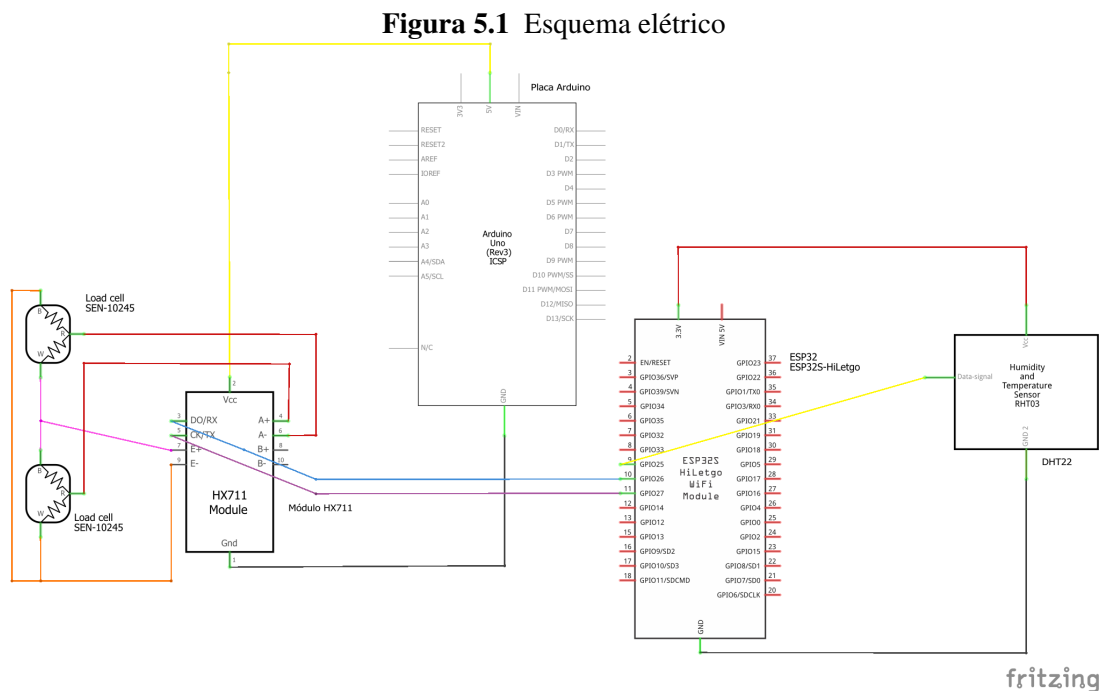
Neste capítulo, apresentam-se detalhadamente os experimentos realizados. Inicialmente, foi abordado o esquema eletrônico do projeto, explicando as conexões entre os sensores e a placa microcontroladora, que foram essenciais para a coleta de dados. E em seguida, foi tratada a configuração do banco de dados no Firebase, detalhando o registro em tempo real dos dados provenientes dos sensores.

Na sequência, foi relatado os primeiros testes com o sensor DHT22 e a placa ESP32, controladora central do projeto, selecionada devido à sua notável versatilidade, eficiência no processamento de dados e sua disponibilidade no mercado. Adicionalmente, foram incorporados sensores especializados de temperatura, umidade e peso, com o propósito de capturar informações cruciais sobre o ambiente das colmeias. Essa abordagem integrada permite uma análise abrangente e precisa dos fatores ambientais que impactam diretamente a saúde e a produtividade das colônias de abelhas.

Por fim, foram realizados os testes em campo, que permitiram verificar o desempenho do sistema em condições reais. Dessa forma, o capítulo busca proporcionar uma visão clara sobre os experimentos realizados e como cada etapa contribuiu para a construção de uma solução prática para o monitoramento apícola.

#### 5.1 Esquema elétrico do projeto

A figura 5.1 representa o esquema elétrico do projeto construído, elaborado no software de código aberto Fritzing do tipo *Computer-Aided Design and Drafting (CAD)* amador para design de hardware eletrônico.



Fonte: Autor

Neste desenho esquemático estão presentes todos os componentes destacados durante o desenvolvimento do protótipo: o sensor DHT22, duas células de carga 50kg, módulo HX711, e as placas ESP32 e Arduino Uno.

Destaque válido a conexão entre as duas *load cells*. Cada uma consiste numa meia ponte de wheatstone, portanto, a fim de obter uma ponte elétrica completa, houve a conexão dos fios brancos com os fios pretos dos sensores (decisão essa que será melhor explicada na seção 5.3.3 deste documento).

Outro destaque importante é justamente quanto a utilização da placa Arduino e da placa ESP32. Pode-se perceber que o Arduino funciona apenas como um "gerador" de energia, responsável por alimentar o módulo HX711 por este funcionar a 5 volts de energia. Já o ESP32 é o meio de comunicação, via Wifi, com o banco de dados Firebase, tendo assim todas as saídas de dados dos sensores conectados a ele, além de ser a responsável pela alimentação elétrica do DHT22.

## 5.2 Conexão com firebase

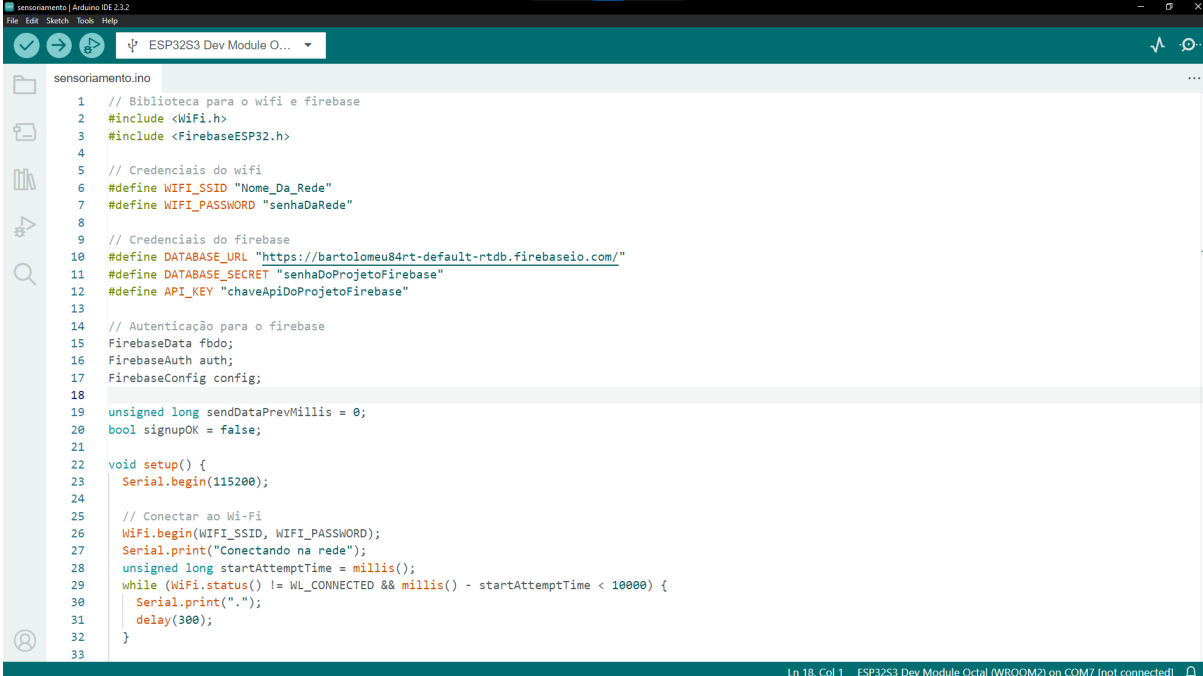
O Firebase é uma plataforma desenvolvida pelo Google que oferece uma gama de serviços voltados para o desenvolvimento de aplicativos web e móveis. Uma das grandes vantagens do Firebase é a sua capacidade de fornecer armazenamento em tempo real, permitindo que os dados sejam sincronizados instantaneamente entre o dispositivo de coleta (como sensores conectados a uma placa microcontroladora) e a nuvem.

Ele é um *Backend as a Service (BaaS)*, possibilitando conectar, no contexto da pesquisa,

a placa ESP32 e os sensores à nuvem. Isso é especialmente útil em projetos de Internet das Coisas (IoT), onde é necessário que os dados sejam acessíveis de qualquer lugar e a qualquer momento desde que haja conexão com a internet. Além disso, o Firebase proporciona uma interface fácil de usar, com suporte para diversas linguagens de programação, o que simplifica a integração com o projeto.

A segurança também é um ponto forte, já que o Firebase permite a implementação de regras de acesso aos dados, garantindo que somente usuários autorizados possam visualizar ou modificar as informações. Os dados são guardados no modelo de *realtime database*, num banco de dados não relacional e permite uma boa transmissão de dados, sendo compatível com diferentes plataformas, como Android, Unit, IOS e Flutter.

Figura 5.2 Código para firebase



```
sensoriamento.ino
1 // Biblioteca para o wifi e firebase
2 #include <WiFi.h>
3 #include <FirebaseESP32.h>
4
5 // Credenciais do wifi
6 #define WIFI_SSID "Nome_Da_Rede"
7 #define WIFI_PASSWORD "senhaDaRede"
8
9 // Credenciais do firebase
10 #define DATABASE_URL "https://bartolomeu84rt-default-rtdb.firebaseio.com/"
11 #define DATABASE_SECRET "senhaDoProjetoFirebase"
12 #define API_KEY "chaveApiDoProjetoFirebase"
13
14 // Autenticação para o firebase
15 FirebaseData fbdo;
16 FirebaseAuth auth;
17 FirebaseConfig config;
18
19 unsigned long sendDataPrevMillis = 0;
20 bool signupOK = false;
21
22 void setup() {
23   Serial.begin(115200);
24
25   // Conectar ao Wi-Fi
26   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
27   Serial.print("Conectando na rede");
28   unsigned long startAttemptTime = millis();
29   while (WiFi.status() != WL_CONNECTED && millis() - startAttemptTime < 10000) {
30     Serial.print(".");
31     delay(300);
32   }
33 }
```

Fonte: Autor

Figura 5.3 Código para firebase

```
33
34 // Caso falhe a conexão
35 if (WiFi.status() != WL_CONNECTED) {
36   Serial.println("Falha na conexão com a rede Wi-Fi!");
37   return;
38 }
39
40 Serial.println();
41 Serial.print("Conectado com IP: ");
42 Serial.println(WiFi.localIP());
43
44 // Configuração do Firebase
45 config.api_key = API_KEY;
46 config.database_url = DATABASE_URL;
47
48 if (Firebase.signUp(&config, &auth, "", "")){
49   Serial.println("Entrou no firebase!");
50   signupOK = true;
51 } else {
52   Serial.printf("%s\n", config.signer.signupError.message.c_str());
53 }
54
55 Firebase.begin(&config, &auth);
56 Firebase.reconnectWiFi(true);
57 void loop() {
58   // put your main code here, to run repeatedly:
59 }
```

Fonte: Autor

As figuras 5.2 e 5.3 mostram as linhas de código necessárias para a conexão com a plataforma Firebase, usada no projeto. Para que o código funcione corretamente, é essencial instalar a "board" do ESP32 e as bibliotecas do Firebase na IDE do Arduino. Isso garante que a placa microcontroladora possa se comunicar com o banco de dados do Firebase, permitindo o envio e recebimento de dados dos sensores em tempo real. A instalação dessas bibliotecas é fundamental para a integração eficiente entre o hardware e o serviço em nuvem, proporcionando uma base sólida para o desenvolvimento do projeto. A seguir, uma explicação mais detalhada.

- `#include <WiFi.h>`: Após baixar a biblioteca WiFi, essa linha inclui a biblioteca no código. Ela permite que a placa ESP32 se conecte a redes Wi-Fi, possibilitando comunicação e transferência de dados pela internet.
- `#include <FirebaseESP32.h>`: Refere-se à biblioteca específica para integrar o ESP32 com o Firebase, facilitando operações como leitura e escrita no banco de dados em tempo real do Firebase.
- `WIFI_SSID` e `WIFI_PASSWORD`: Definem o nome da rede Wi-Fi e a senha à qual o ESP32 tentará se conectar. É necessário substituir pelas credenciais da rede que será utilizada.
- `DATABASE_URL`: Uniform Resource Locator (URL) do seu banco de dados Firebase Realtime Database. Deve ser configurado conforme o endereço do seu projeto no Firebase.

- `DATABASE_SECRET` e `API_KEY`: Credenciais de autenticação para acessar o Firebase. Essas chaves permitem que o ESP32 se autentique e interaja com o banco de dados de forma segura.
- `FirestoreData fbdo`: Cria um objeto que gerencia as operações de envio e recebimento de dados com o Firebase.
- `FirestoreAuth auth`: Objeto responsável pela autenticação com o Firebase, utilizando as credenciais fornecidas.
- `FirestoreConfig config`: Objeto que armazena as configurações do Firebase, incluindo a API Key e o Database URL.
- `unsigned long sendDataPrevMillis = 0`: Variável usada para controlar o intervalo de tempo entre os envios de dados, ajudando a evitar sobrecarga na comunicação.
- `bool signupOK = false`: Variável booleana que indica se a autenticação com o Firebase foi realizada com sucesso.
- `Serial.begin(115200)`: Inicia a comunicação serial com uma taxa de transmissão de 115200 bps, permitindo o monitoramento e depuração através do monitor serial.
- `WiFi.begin(WIFI_SSID, WIFI_PASSWORD)`: Inicia o processo de conexão à rede Wi-Fi usando as credenciais fornecidas.
- O `"while`  
em verde na linha 29 é um loop que verifica continuamente o status da conexão, tentando conectar por até 10 segundos. Durante o processo, imprime pontos no monitor serial para indicar a tentativa de conexão.
- O bloco que começa na linha 35 verifica se a conexão com o Wi-Fi foi bem-sucedida. O método `WiFi.status()` retorna o status da conexão. `WL_CONNECTED` é um estado que indica que o dispositivo está conectado a uma rede Wi-Fi. Se a conexão falhar, ou seja, se o status não for `WL_CONNECTED`, uma mensagem de erro é impressa no monitor serial: "Falha na conexão com a rede Wi-Fi!"  
. Em seguida, o comando `return;` faz com que a função `setup()` seja encerrada prematuramente, impedindo que o restante do código seja executado sem uma conexão de rede estabelecida.
- Após a verificação de conexão, `Serial.println();` é usado para imprimir uma linha em branco no monitor serial. `Serial.print("Conectado com IP: ");` imprime a mensagem indicando que a conexão foi bem-sucedida e que o IP do dispositivo será exibido a seguir. `WiFi.localIP()` retorna o endereço IP local atribuído ao dispositivo após

a conexão ao Wi-Fi. `Serial.println(WiFi.localIP());` imprime esse endereço IP no monitor serial.

- Na linha 45, há `config.api_key = API_KEY;` Configura a chave da API do Firebase, necessária para autenticação e acesso ao banco de dados. `config.database_url = DATABASE_URL;` Define a URL do banco de dados Firebase onde os dados serão armazenados e acessados.
- `Firebase.signUp(&config, &auth, , )`: Tenta realizar a autenticação anônima no Firebase usando as configurações previamente definidas (`config` e `auth`).
  - `&config`: Referência ao objeto de configuração do Firebase.
  - `&auth`: Referência ao objeto de autenticação.
  - Se a autenticação for bem-sucedida (`if`), a mensagem "Entrou no firebase!" é impressa no monitor serial, e a variável `signupOK` é definida como `true`, indicando sucesso na autenticação.
  - Caso a autenticação falhe (`else`), o código exibe a mensagem de erro correspondente utilizando `Serial.printf`, que formata e imprime a mensagem de erro.
- `Firebase.begin(&config, &auth)`: Inicializa a conexão com o Firebase utilizando as configurações e autenticações definidas. Isso é necessário para que o dispositivo comece a interagir com o Firebase.
- `Firebase.reconnectWiFi(true)`: Habilita a reconexão automática ao Wi-Fi caso a conexão seja perdida. Isso garante que o dispositivo se reconecte automaticamente à rede sem a necessidade de intervenção manual.

### 5.3 Primeiros testes

Nas seções abaixo serão mostrados os testes realizados com os componentes eletrônicos do projeto, de forma individual.

Para os experimentos feitos com os sensores (DHT22 e a célula de carga 50kg), foi utilizada a placa Arduino como microcontrolador central. O objetivo era apenas averiguar o funcionamento esperado dos sensores, por isso a utilização desta placa.

Na conclusão do protótipo, foi usada a placa ESP32, justamente por esta prover a capacidade de processamento necessária para os dados fornecido pelos sensores, e realizar a comunicação e envio das informações ao banco de dados Firebase.

### 5.3.1 Teste com DHT22

Foram realizados os primeiros testes em um dos sensores que serão usados no protótipo, o DHT22. O experimento foi feito para averiguar o funcionamento do sensor, bem como a confeccionar o código responsável pelo seu funcionamento, conforme a figura 5.4.

**Figura 5.4** Código Sensor DHT22

```
DHT22.ino
1  #include <DHT.h>; //INCLUSÃO DE BIBLIOTECA
2
3  #define DHTPIN 7 //PINO DIGITAL UTILIZADO PELO DHT22
4  #define DHTTYPE DHT22 //DEFINE O MODELO DO SENSOR (DHT22 / AM2302)
5
6  DHT dht(DHTPIN, DHTTYPE); //PASSA OS PARÂMETROS PARA A FUNÇÃO
7
8  void setup(){
9      Serial.begin(9600); //INICIALIZA A SERIAL
10     dht.begin(); //INICIALIZA A FUNÇÃO
11     delay(2000); //INTERVALO DE 2 SEGUNDO ANTES DE INICIAR
12 }
13 void loop(){
14     Serial.print("Umidade: "); //IMPRIME O TEXTO NA SERIAL
15     Serial.print(dht.readHumidity()); //IMPRIME NA SERIAL O VALOR DE UMIDADE MEDIDO
16     Serial.print("%"); //IMPRIME O TEXTO NA SERIAL
17     Serial.print(" / Temperatura: "); //IMPRIME O TEXTO NA SERIAL
18     Serial.print(dht.readTemperature(), 0); //IMPRIME NA SERIAL O VALOR DE UMIDADE MEDIDO E REMOVE A PARTE DECIMAL
19     Serial.println("°C"); //IMPRIME O TEXTO NA SERIAL
20     delay(2000); //INTERVALO DE 2 SEGUNDOS * NÃO DIMINUIR ESSE VALOR
21 }
22
```

Fonte: Autor

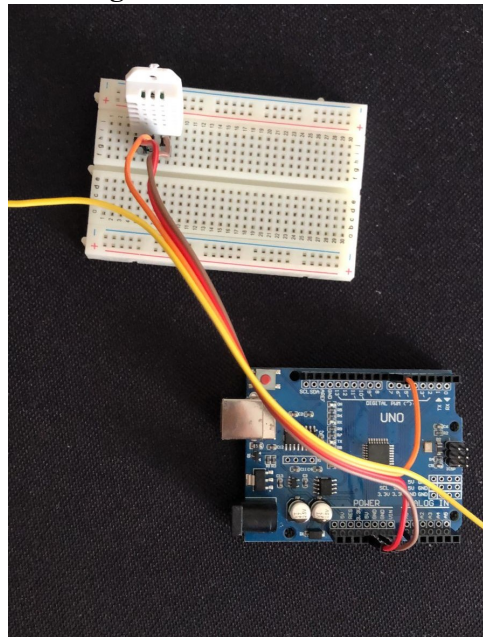
O código apresentado na figura 5.4 é um exemplo de programa com linguagem C++, projetado para ser executado na plataforma Arduino. O programa realiza leituras periódicas do sensor de temperatura e umidade (DHT22), utilizando a biblioteca DHT, e exibe os resultados no monitor serial.

A seguir, uma explicação do funcionamento de cada linha contida no código:

- "include <DHT.h>": Esta linha permite a inclusão da biblioteca DHT no programa. Para isso, foi necessário baixar um arquivo "DHT\_sensor\_library.zip" e incluí-lo na plataforma;
- "define DHTPIN 7": Define o pino digital ao qual o sensor DHT22 está conectado no Arduino como sendo o pino 7.
- "define DHTTYPE DHT22": Define o tipo de sensor como DHT22, visto que a biblioteca utilizada serve tanto para o funcionamento do DHT11 quanto ao do DHT22.
- "DHT dht(DHTPIN, DHTTYPE)": Cria um objeto dht do tipo DHT, associado ao pino definido (DHTPIN) e ao tipo de sensor (DHTTYPE).
- "void setup() ": Define as configurações iniciais
- "Serial.begin(9600)": Inicializa a comunicação serial a uma taxa de 9600 baud (número de vezes por segundo que um sinal de comunicação serial muda estados).

- "dht.begin()": Inicializa o objeto DHT para iniciar a comunicação com o sensor.
- "delay(2000)": Aguarda 2 segundos antes de iniciar a execução principal do programa.
- "void loop() ": Define as instruções que serão executadas repetidamente.
- "Serial.print() e Serial.println()": Instruções para imprimir informações na porta serial, que pode ser usada para monitorar e depurar o programa.
- "dht.readHumidity()": Lê e imprime a umidade medida pelo sensor.
- "dht.readTemperature()": Lê e imprime a temperatura medida pelo sensor.
- "delay(2000)": Introduce um atraso de 2 segundos entre as leituras consecutivas.

**Figura 5.5** Circuito DHT22



Fonte: Autor

Na imagem 5.5 pode-se visualizar a montagem do circuito que permite o funcionamento do sensor DHT22.

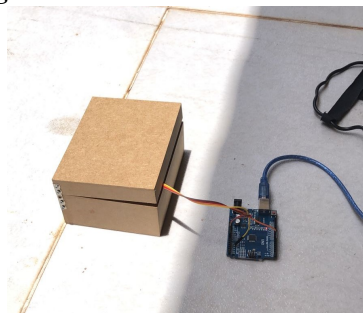
Com o auxílio de uma protoboard, o pino VCC do sensor é conectado à entrada de 5 volts de energia da placa Arduino, por meio do jumper vermelho; o pino GND é conectado, pelo jumper marrom, à entrada GND da placa; e por fim o pino de DATA é ligado à entrada digital 7 do microcontrolador, por um jumper laranja. Todos os jumpers utilizados na montagem são do tipo macho-macho.

Os experimentos que foram realizados com esse sensor consistiam em observar e avaliar a diferença na medição dos valores de temperatura e umidade em 3 caixas de materiais diferentes (madeira, plástico e papelão), quando expostos ao sol.

O sensor, quando colocado em cada caixa individualmente, foi exposto ao sol por 6 minutos (o teste como um todo durou de 11:54 à 12:26), com tempos de intervalo de repouso entre cada teste de aproximadamente 7 minutos. Todas as leituras de temperatura e umidade foram visualizadas diretamente do monitor serial da plataforma Arduino, no computador pessoal de um dos autores.

Na caixa de madeira, com a temperatura inicialmente em 32°C e umidade em 50%. Com o teste finalizado nessa caixa, observou-se uma variação de 3°C (32° a 35°C) de temperatura, e uma variação entre 54% e 60% na umidade.

**Figura 5.6** Teste na caixa de madeira



Fonte: Autor

Com a caixa de plástico, antes de iniciar o teste a temperatura constava como 34°C e a umidade em 45,5%. Após a finalização do teste, a temperatura apresentou variação de 4°C (34° a 38°C) enquanto a umidade apresentou queda após o início do teste, chegando ao final em aproximados 34,5%.

**Figura 5.7** Teste na caixa de plástico



Fonte: Autor

Por fim, na caixa de papelão, com temperatura inicial de 34°C e umidade de 42%, com a conclusão dos 6 minutos de experimento, a temperatura variou em 6°C (34° a 40°C) enquanto a umidade baixou para casa de 33,8%.

**Figura 5.8** Teste na caixa de papelão

Fonte: Autor

Os resultados dos testes foram extremamente significativos para o andamento deste trabalho e o prosseguindo na construção do protótipo, pois além de notar que a temperatura interna em certos tipos de materiais possuem diferença, o sensor DHT22 se mostrou útil e eficaz na coleta das variáveis exigidas.

### 5.3.2 Teste com ESP32

Também foi conduzido um teste com a placa microcontroladora ESP32 e o sensor LM35 (um sensor de temperatura). Este sensor se destaca pela simplicidade de uso e sua precisão. Já a placa é equipada com chip Wi-Fi, o que possibilita uma comunicação mais simples com o servidor para transmissão de informações. A montagem envolveu a conexão do sensor à placa microcontroladora ESP32, e para fornecer energia foi usado uma fonte de 5 volts.

Ainda é esperado a parte prática de leitura de temperatura, para isso foi usada a tecnologia *Message Queuing Telemetry (MQTT)*, a placa se conecta ao servidor local, e manda as informações coletadas por meio da internet, que podem ser acessadas caso o dispositivo esteja conectado na mesma rede wifi. Para isso também é necessário um computador pessoal, que será o responsável por hospedar o server local. Essa etapa prática não apenas validou a escolha da tecnologia de transmissão, mas também ressaltou a importância da comunicação estável para um monitoramento preciso.

O processo referente à conexão ao Broker (conexão à um software que permite aceder a desktops e aplicações hospedados em servidores) corresponde ao seguinte trecho de código:

**Figura 5.9** Código para conexão

```
10 // MQTT client
11 WiFiClient wifiClient;
12 PubSubClient mqttClient(wifiClient);
13 char *mqttServer = "mqtt_broker_address";
14 int mqttPort = mqtt_broker_gate;
15
16
17 //LM35 variable declarations
18 int LM35_Raw_Sensor1 = 0;
19 float temperature = 0.0;
20 char tempString[8];
21 float Voltage = 0.0;
22
23 //Variable declarations
24 long last_time = 0;
25 bool tempChange = false;
26
27 //Timer declaration
28 hw_timer_t * timer = NULL;
29
30 /* Setup Function for WiFi Connection */
31 void connectToWiFi() {
32     Serial.print("Ligação a");
33
34     WiFi.begin(ssid, password);
35     Serial.println(ssid);
36     while (WiFi.status() != WL_CONNECTED) {
37         Serial.print(".");
38         delay(500);
39     }
40     Serial.print("Conectado.");
41 }
42
43 /* Setup Function for MQTT Communication */
44 void setupMQTT() {
45     mqttClient.setServer(mqttServer, mqttPort);
46     // define a função de retorno de chamada
47     mqttClient.setCallback(callback);
48 }
```

Fonte: Autor

A seguir, uma lista de comentários com o funcionamento de cada linha contida no código:

- MQTT client: Declara o cliente MQTT e a variável Wi-Fi.
- WiFiClient wifiClient;: Inicializa o cliente Wi-Fi, que será utilizado para a conexão.
- PubSubClient mqttClient(wifiClient);: Configura o cliente MQTT utilizando a conexão Wi-Fi.
- char \*mqttServer = "mqtt\_broker\_address";: Define o endereço do servidor MQTT.
- int mqttPort = mqtt\_broker\_gate;: Define a porta usada pelo servidor MQTT.
- LM35 variable declarations: Declaração das variáveis utilizadas para o sensor LM35.
- int LM35\_Raw\_Sensor1 = 0;: Inicializa a variável para armazenar o valor bruto do sensor LM35.
- float temperature = 0.0;: Inicializa a variável para armazenar a temperatura calculada.
- char tempString[8];: Inicializa a variável para armazenar a temperatura em formato de string.
- float Voltage = 0.0;: Inicializa a variável para armazenar a tensão medida pelo sensor.
- Variable declarations: Declaração de variáveis auxiliares.

- `long last_time = 0;`: Inicializa a variável para armazenar o tempo da última execução.
- `bool tempChange = false;`: Inicializa a variável booleana para indicar mudança de temperatura.
- `Timer declaration:` Declaração do timer de hardware.
- `hw_timer_t * timer = NULL;`: Inicializa o ponteiro para o timer de hardware.
- `Setup Function for WiFi Connection:` Função de configuração para a conexão Wi-Fi.
- `void connectToWiFi() {` Início da função para conectar ao Wi-Fi.
- `Serial.print("Ligação a");`: Imprime "Ligação a" no monitor serial.
- `WiFi.begin(ssid, password);`: Inicia a conexão Wi-Fi usando as credenciais fornecidas.
- `Serial.println(ssid);`: Imprime o SSID da rede no monitor serial.
- `while (WiFi.status() != WL_CONNECTED) {` Verifica se a conexão Wi-Fi foi estabelecida.
- `Serial.print(".");`: Imprime um ponto no monitor serial enquanto tenta conectar.
- `delay(500);`: Aguarda 500 milissegundos antes de tentar novamente.
- `}`: Fecha o loop while.
- `Serial.print("Conectado.");`: Imprime "Conectado." no monitor serial quando a conexão é bem-sucedida.
- `Setup Function for MQTT Communication:` Função de configuração para a comunicação MQTT.
- `void setupMQTT() {` Início da função para configurar o MQTT.
- `mqttClient.setServer(mqttServer, mqttPort);`: Define o servidor e a porta para o cliente MQTT.
- `// define a função de retorno de chamada:` Comentário indicando a função de callback.
- `mqttClient.setCallback(callback);`: Define a função de *callback* (retorno) para o cliente MQTT.

Consoante com os testes anteriores, foi conduzido um experimento do que foi desenvolvido. A metodologia adotada incluiu a posição estratégica do sensor LM35 para capturar variações térmicas, enquanto a placa ESP32 coletava e transmitia os dados em tempo real. Os resultados preliminares destacaram a eficácia na captura de dados e a latência na transmissão para o servidor local. Essas observações são essenciais para aprimorar o projeto antes da implementação nos apiários reais, garantindo uma solução robusta para os apicultores.

**Figura 5.10** Teste da placa ESP32



Fonte: Autor

Para ilustrar essa fase de teste, incluiu-se uma foto da caixa de sapato, com dimensões 33 x 19 x 12 cm (figura 5.10), visualmente destacando o ambiente controlado. Essas informações fornecem *insights* (informações) valiosos para o desenvolvimento contínuo do projeto, assegurando sua confiabilidade e eficiência nos apiários reais.

### 5.3.3 Teste com a célula de carga 50kg

A equipe realizou experimentos com diferentes objetos para verificar o funcionamento adequado da célula de carga 50kg, com seu respectivo código de execução.

O código utilizado foi programado em linguagem C++ na plataforma Arduino IDE, assim como no DHT22. Disponibilizado pela biblioteca utilizada, HX711\_ADC library, na aba "Examples" do software, as linhas de código são responsáveis por permitir a calibração das células, bem como a leitura periódica dos valores captados.

Para o projeto, foi confeccionado uma balança com duas *load cells* (conectadas uma a outra para funcionarem como apenas uma), a qual consiste numa placa de madeira, cortada em uma máquina *Computer Numeric Control (CNC)* de corte a laser, do Instituto Federal de Brasília (IFB), como base, pensada com as dimensões para encaixar todo o circuito do protótipo.

**Figura 5.11** Placa de madeira base da balança

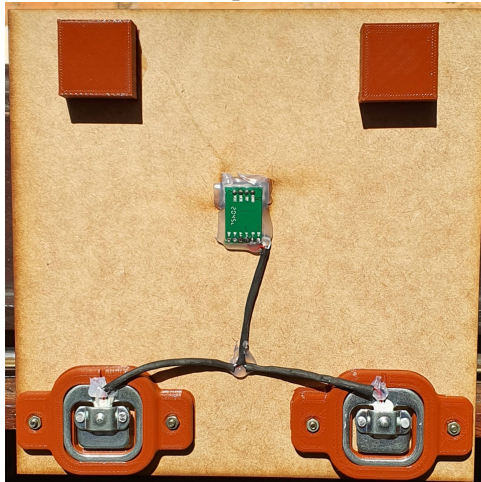
Fonte: Autor

Os sensores foram afixados a placa com auxílio de uma estrutura confeccionada na impressora 3D do laboratório Maker do IFB (como visto na imagem 5.12). Além disso, com o mesmo material, foram criados alguns pequenos blocos para nivelar a balança e deixar da melhor forma possível para a medição dos pesos.

**Figura 5.12** Suporte 3D

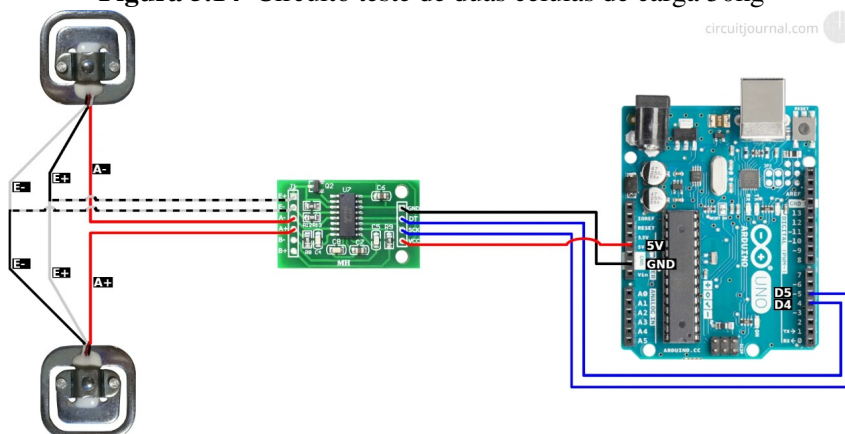
Fonte: Autor

De acordo com a figura 5.13, pode-se notar que as células de carga estão conectadas uma a outra, funcionando como um único sensor.

**Figura 5.13** Visão da parte de baixo da balança

Fonte: Autor

Os fios vermelhos de cada *strain gauge* conectam-se aos pinos A+ e A- do módulo HX711. Isso foi feito porque cada *load cell* consiste em uma meia ponte de wheatstone, então para conseguir uma ponte completa, os fios branco das células foram soldados aos fios preto da outra célula, e essas uniões foram conectadas aos pinos E+ e E- do módulo. Em seguida, os pinos VCC e GND do HX711 são conectados as entradas 5V e GND da placa Arduino, enquanto os pinos DT e SCK são ligados as portas digitais 4 e 5, respectivamente.

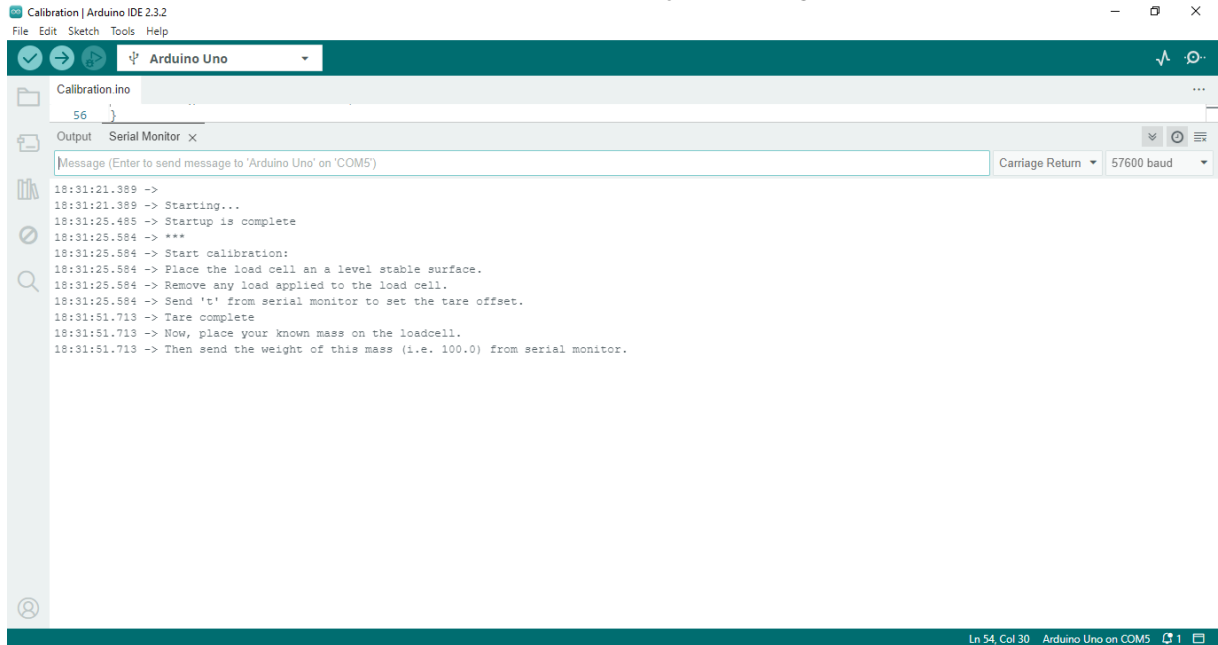
**Figura 5.14** Circuito teste de duas células de carga 50kg

Fonte: circuitjournal.com

Quanto aos testes executados, tinham o objetivo de avaliar e conferir o funcionamento adequado das células de carga, para isso foram utilizados 3 objetos de pesos distintos.

Ao completar o upload do código na placa, no monitor serial é necessário zerar a balança (sem que haja nenhum objeto sobre ela), e posteriormente informar algum peso conhecido para obter o fator de calibração das células.

Figura 5.15 Inicialização do código

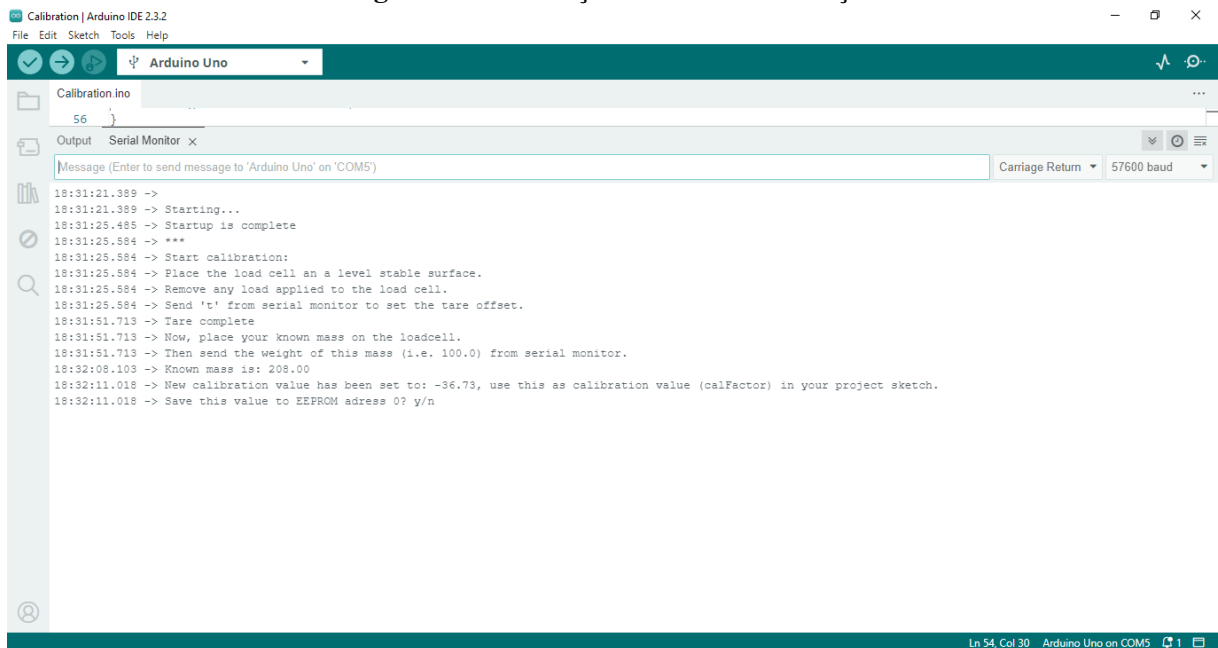


```
Calibration | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
Calibration.ino
56
Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM5') Carriage Return 57600 baud
18:31:21.389 ->
18:31:21.389 -> Starting...
18:31:25.485 -> Startup is complete
18:31:25.584 -> ***
18:31:25.584 -> Start calibration:
18:31:25.584 -> Place the load cell an a level stable surface.
18:31:25.584 -> Remove any load applied to the load cell.
18:31:25.584 -> Send 't' from serial monitor to set the tare offset.
18:31:51.713 -> Tare complete
18:31:51.713 -> Now, place your known mass on the loadcell.
18:31:51.713 -> Then send the weight of this mass (i.e. 100.0) from serial monitor.
Ln 54, Col 30 Arduino Uno on COM5
```

Fonte: Autor

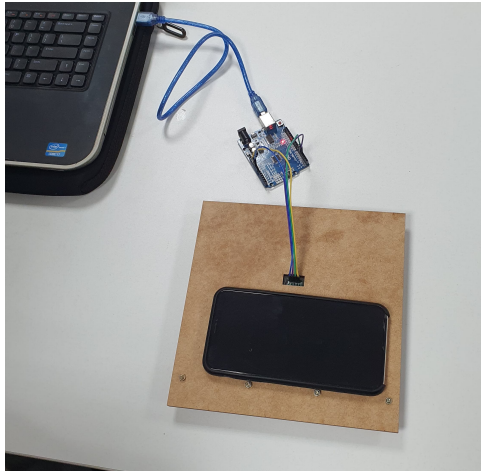
Para o primeiro teste (figura 5.17), foi utilizado o celular de um dos integrantes da equipe, que apresentou um valor próximo a 208 gramas. Esse valor foi informado inicialmente ao código e assim foi gerado o fator de calibração (figura 5.16).

Figura 5.16 Definição do fator de calibração



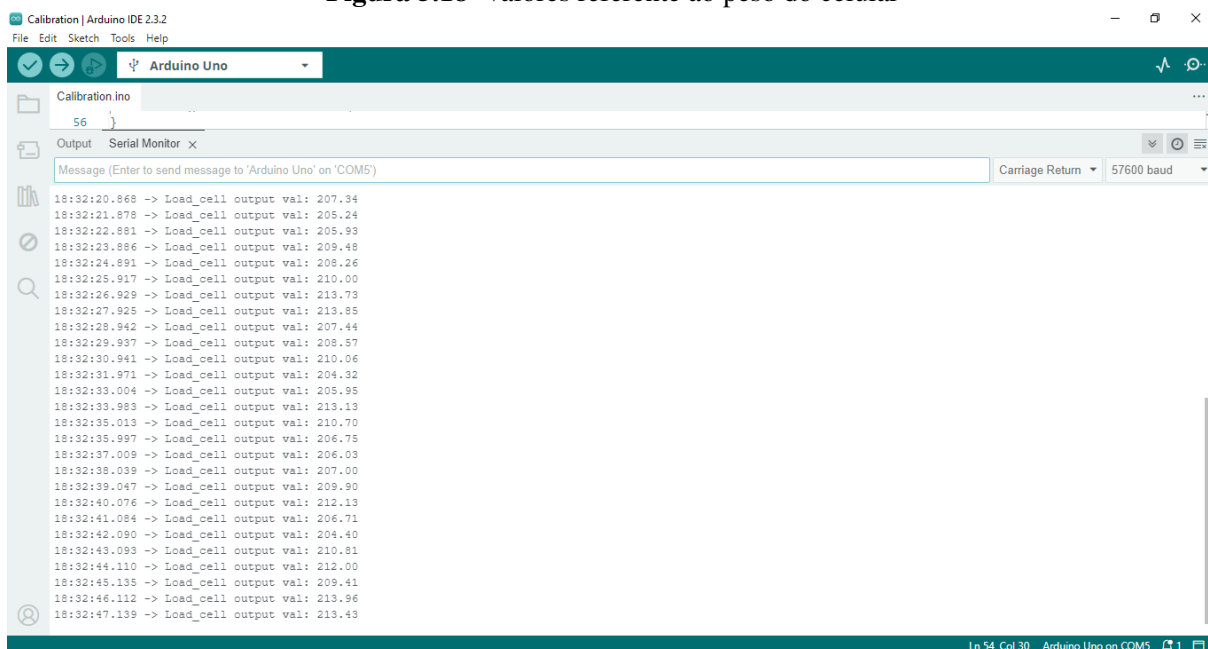
```
Calibration | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
Calibration.ino
56
Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM5') Carriage Return 57600 baud
18:31:21.389 ->
18:31:21.389 -> Starting...
18:31:25.485 -> Startup is complete
18:31:25.584 -> ***
18:31:25.584 -> Start calibration:
18:31:25.584 -> Place the load cell an a level stable surface.
18:31:25.584 -> Remove any load applied to the load cell.
18:31:25.584 -> Send 't' from serial monitor to set the tare offset.
18:31:51.713 -> Tare complete
18:31:51.713 -> Now, place your known mass on the loadcell.
18:31:51.713 -> Then send the weight of this mass (i.e. 100.0) from serial monitor.
18:32:08.103 -> Known mass is: 208.00
18:32:11.018 -> New calibration value has been set to: -36.73, use this as calibration value (calFactor) in your project sketch.
18:32:11.018 -> Save this value to EEPROM address 0? y/n
Ln 54, Col 30 Arduino Uno on COM5
```

Fonte: Autor

**Figura 5.17** Teste realizado com o celular

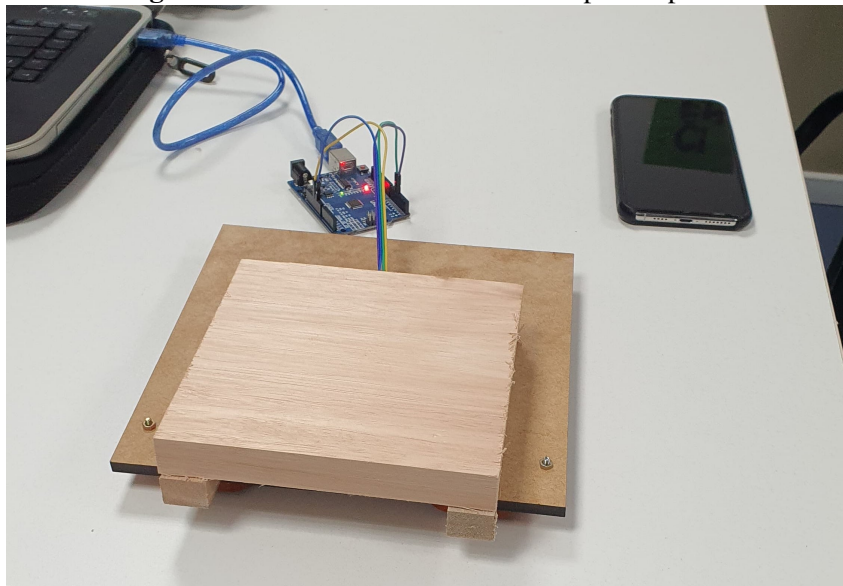
Fonte: Autor

Os valores produzidos pela deformação das duas células variaram entre 204 gramas a 213 gramas, como apresentado na imagem 5.18.

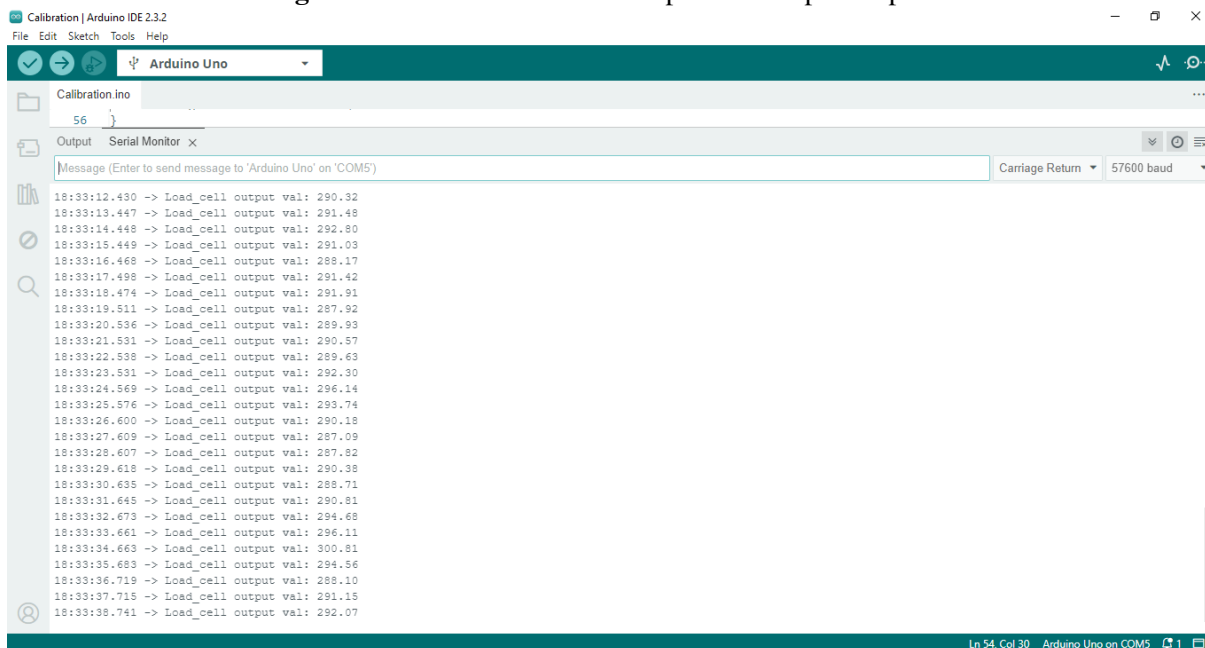
**Figura 5.18** Valores referente ao peso do celular

Fonte: Autor

Para o segundo teste, utilizou-se a tampa do apiário comprado pela equipe (figura 5.19). O peso desse objeto é de 290 gramas. Na captação pelo sensor, os dados variaram entre 287 gramas a 300 gramas (imagem 5.20).

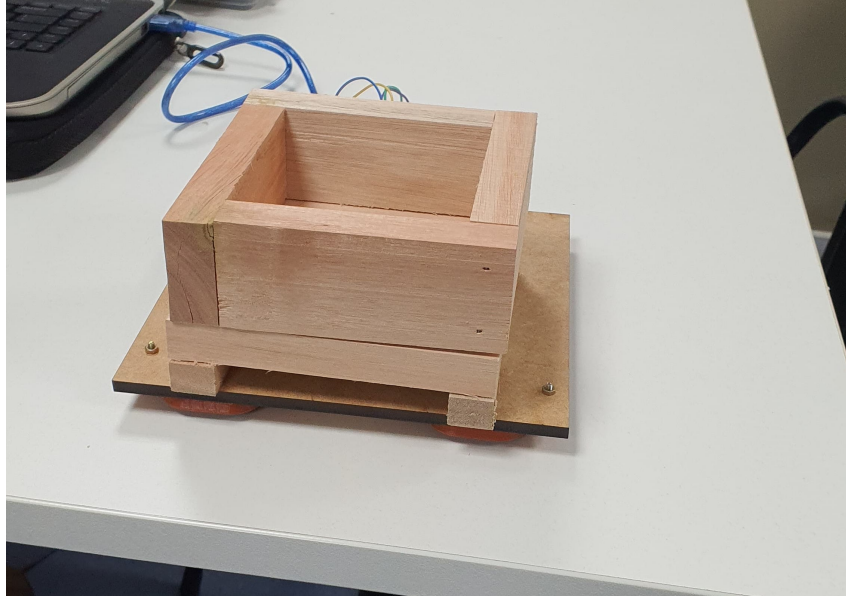
**Figura 5.19** Teste realizado com a tampa do apiário

Fonte: Autor

**Figura 5.20** Valores referente ao peso da tampa do apiário

Fonte: Autor

No terceiro teste realizado, foi utilizado a base do apiário adquirido (figura 5.21). Seu valor de peso é de 486 gramas, porém esse dado variou entre 479 e 491 gramas (imagem 5.22).

**Figura 5.21** Teste realizado com a base do apiário

Fonte: Autor

**Figura 5.22** Valores referente ao peso da base do apiário

```
Calibration | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
Calibration.ino
56
Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM5')
Carriage Return 57600 baud
18:33:58.935 -> Load_cell output val: 495.28
18:33:59.962 -> Load_cell output val: 489.10
18:34:00.956 -> Load_cell output val: 487.67
18:34:01.972 -> Load_cell output val: 485.94
18:34:03.001 -> Load_cell output val: 486.20
18:34:03.991 -> Load_cell output val: 487.67
18:34:05.033 -> Load_cell output val: 483.47
18:34:06.030 -> Load_cell output val: 485.24
18:34:07.023 -> Load_cell output val: 489.61
18:34:08.061 -> Load_cell output val: 491.65
18:34:09.058 -> Load_cell output val: 488.68
18:34:10.050 -> Load_cell output val: 487.57
18:34:11.081 -> Load_cell output val: 489.43
18:34:12.107 -> Load_cell output val: 487.53
18:34:13.127 -> Load_cell output val: 481.93
18:34:14.104 -> Load_cell output val: 479.73
18:34:15.123 -> Load_cell output val: 480.78
18:34:16.134 -> Load_cell output val: 484.52
18:34:17.160 -> Load_cell output val: 487.68
18:34:18.161 -> Load_cell output val: 486.87
18:34:19.166 -> Load_cell output val: 484.63
18:34:20.204 -> Load_cell output val: 482.53
18:34:21.193 -> Load_cell output val: 483.49
18:34:22.194 -> Load_cell output val: 485.17
18:34:23.221 -> Load_cell output val: 486.81
18:34:24.216 -> Load_cell output val: 484.31
18:34:25.236 -> Load_cell output val: 484.50
Ln 54, Col 30 Arduino Uno on COM5
```

Fonte: Autor

Com o desenvolvimento do circuito, a equipe esperava que os resultados obtidos fossem mais precisos. Contudo, por conta do tempo limitado para a finalização do projeto, foi dado continuidade a montagem. Por mais que as variações sejam aproximadamente de 1 a 2%, o momento em que a colmeia de teste se encontrava permitiu à equipe continuar com a execução do projeto (como será visto na seção 5.6). Para melhorias futuras, espera-se configurar as *load cells* para produzirem um valor mais exato.

### 5.3.4 Testes com sensores e firebase

Após estudos, realização e análises feitas nos testes anteriores, foram conduzidos novos testes com o sensor DHT22, que mede a temperatura e umidade relativa do ar, e a célula de carga, que mede a massa de um objeto. Esses sensores estão conectados à placa microcontroladora ESP32, a mesma manda as informações lidas por esses sensores para o banco de dados no Firebase. Para que haja uma conexão com a internet deve ser digitado no código o nome da rede Wi-Fi que a placa microcontroladora irá se conectar, e a sua senha, para realizar a conexão. Para estabelecer uma conexão com o banco de dados é necessário informar, no código, a *Application Programming Interface (API)*, a URL e senha do projeto que está na plataforma. Então, após o código ser enviado para a placa por meio da Arduino IDE, é possível conectar a uma fonte de energia, em uma tomada ou um computador por exemplo.

Figura 5.23 Alteração no banco de dados do firebase



Fonte: Autor

Então, se a conexão for estabelecida, será possível ver os dados sendo alterados na plataforma do Firebase. A mesma pode ser acessada de qualquer lugar, desde que haja conexão com a internet e que o usuário tenha as credenciais para o acesso. Dessa forma, ambos os sensores estão enviando as informações corretamente para os seus respectivos campos na estrutura não relacional do Firebase. E com a API e informações do banco de dados é possível mostrar esses dados em uma página web ou um aplicativo.

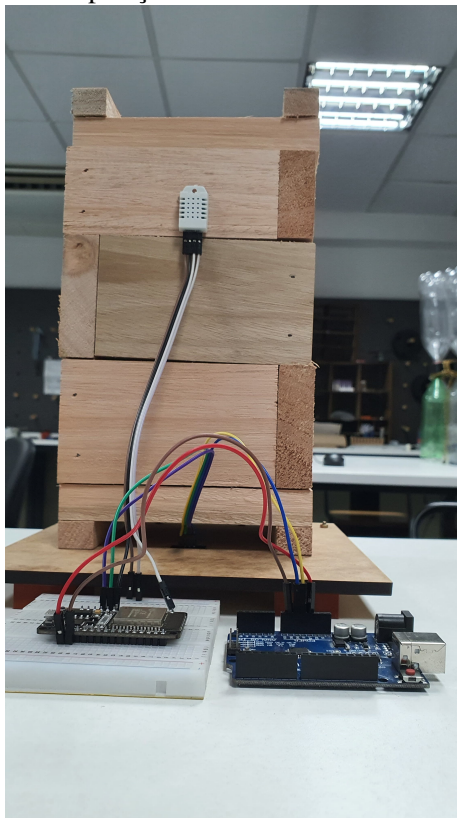
## 5.4 Posicionamento dos sensores

A correta disposição dos sensores dentro da colmeia é fundamental para garantir a precisão das leituras e o bem-estar das abelhas. Os sensores usados são o DHT22, que mede

temperatura e umidade, e a célula de carga, responsável por monitorar o peso da colmeia. O sensor DHT22 deve ser posicionado no interior da colmeia, preferencialmente em uma área central como a área de cria das larvas, onde as abelhas mantêm a temperatura estável. Essa localização estratégica permite que o sensor capture com precisão as condições ambientais internas. A precisão das medições de temperatura e umidade oferecidas pelo DHT22 é um dos seus principais pontos fortes, o que o torna ideal para o monitoramento contínuo das condições climáticas dentro da colmeia. Além disso, seu baixo consumo de energia e design compacto facilitam a instalação em espaços confinados, ele também já apresenta uma pequena alça de plástico, que facilita a montagem no projeto. No entanto, a proximidade com as abelhas pode interferir nas leituras devido ao calor gerado por elas, e o sensor pode ser coberto com cera ao longo do tempo, o que comprometeria a precisão das medições.

A célula de carga, por sua vez, deve ser instalada na base da colmeia, suportando todo o peso da estrutura. Essa posição permite que o sensor registre variações de peso da caixa como um todo, oferecendo dados sobre a produção de mel, a entrada de néctar, e a atividade geral dentro da colmeia. A capacidade da célula de carga de detectar mudanças sutis no peso é essencial para monitorar a produtividade da colmeia e identificar possíveis problemas, como o abandono da colmeia. No entanto, para garantir que as leituras permaneçam precisas, é necessária uma calibração regular da célula de carga, pois a célula de carga pode constantemente oferecer pequenas alterações sem que haja uma mudança real.

**Figura 5.24** Disposição idealizada dos sensores no apiário



Fonte: Autor

## 5.5 Teste contínuo

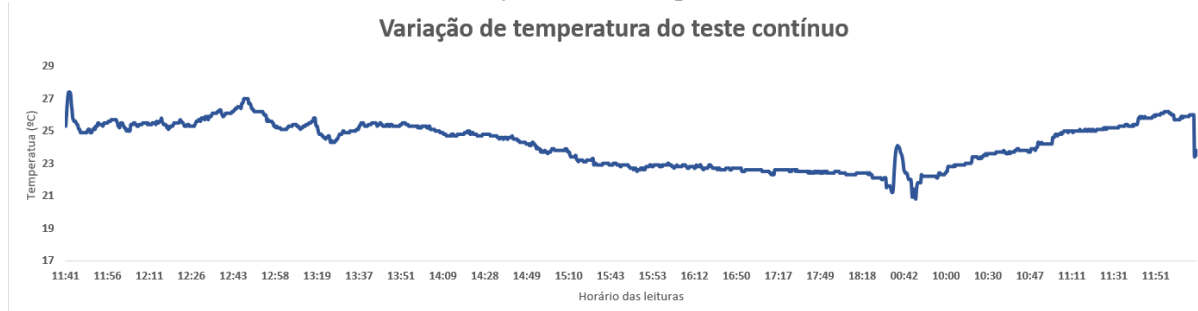
Antes de implementar o sistema em campo, foram realizados testes controlados em laboratório e feito um com duração superior a 24 horas. O objetivo principal foi avaliar a consistência na coleta de dados, a capacidade de transmissão ao longo do tempo e possíveis falhas no sistema.

O teste teve início às 11h40, como mostra na figura 5.25, com o sistema configurado para realizar leituras contínuas e atualizar os dados no banco quando houvesse alguma mudança significativa no sensor. Poucos minutos após o início, foi colocado um peso de cerca de 900 gramas sobre a célula de carga. O peso foi mantido por algumas horas, e às 13h01 foi adicionado mais peso à carga total. Às 18h54, uma parte do peso foi retirada, o que permitiu observar claramente as variações no gráfico gerado. Os picos presentes no gráfico se devem ao manuseio, que levou a uma leitura alta do peso.

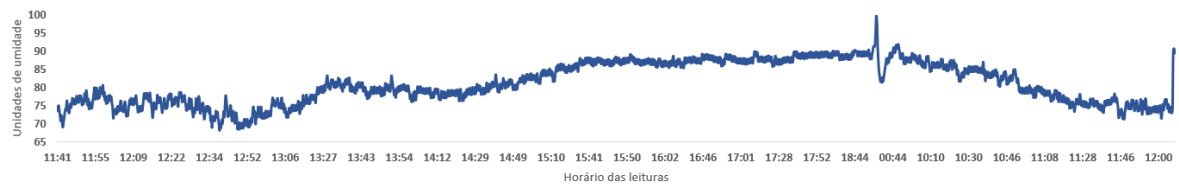
**Figura 5.25** Variação lida do peso durante o teste



**Figura 5.26** Variação lida da temperatura durante o teste



**Figura 5.27** Variação lida da umidade durante o teste  
Variação de umidade do teste contínuo



Além disso, o sistema demonstrou capacidade de captar mudanças ambientais. Durante o teste, houve ocorrência de chuva, o que resultou em uma elevação nos níveis de umidade relativa registrados pelo sensor. A umidade subiu de 76,5 às 12h00 para 90 às 18h55. Esses resultados comprovam a funcionalidade do sistema em monitorar fatores externos como peso, temperatura e umidade de forma contínua e confiável.

## 5.6 Teste em campo

Os testes de campo foram realizados no apiário do apicultor Bruno, que mantém colmeias de abelhas da espécie Jataí (*Tetragonisca angustula*). Este ambiente proporcionou uma oportunidade de avaliar o desempenho do sistema em condições reais e sob interação direta com as colmeias.

**Figura 5.28** Sistema protegido com a caixinha de plástico



Fonte: Autor

Para realizar o teste o apicultor se dispôs a instalar uma prateleira para receber o sistema, ele foi colocado na balança e os componentes em cima. Para proteger o módulo ESP32 e outros

componentes eletrônicos, foi utilizado uma caixa plástica com tampa, garantindo resistência contra chuva, poeira e outros fatores ambientais.

Os dados coletados demonstraram precisão e coerência com as condições do ambiente. Observou-se, por exemplo, uma diferença significativa entre os valores registrados dentro da colmeia e os do ambiente externo, evidenciando o impacto do microclima interno das colmeias na vida das abelhas. Esses resultados reforçam a importância de monitorar diretamente o interior das caixas para compreender melhor os fatores que influenciam a saúde e produtividade das colmeias.

Durante a visita, o apicultor Bruno mostrou interesse nos dados coletados, destacando a utilidade do sistema para o manejo das abelhas e a previsão de problemas nas colmeias. Ele também forneceu feedback valioso sobre o posicionamento dos sensores e a usabilidade do sistema, o que poderá orientar ajustes futuros.

Orientado pelo apicultor, os dados coletados mostraram-se condizentes com o momento da colmeia. Como havia sido recém realizada a transposição da colmeia, inicialmente coletada, para esta nova caixa utilizada na pesquisa, as abelhas estavam na fase de reconstrução de seu habitat. Portanto, é natural que não fosse captada nenhuma alteração significativa das condições ambientais, principalmente quanto ao peso, notando uma diferença mínima entre os diferentes valores coletados. Os dados foram coletados ao longo de 1 semana e meia, sendo incapaz de medir alterações de peso significativas, visto que a produção de mel, por exemplo, é um processo de meses.

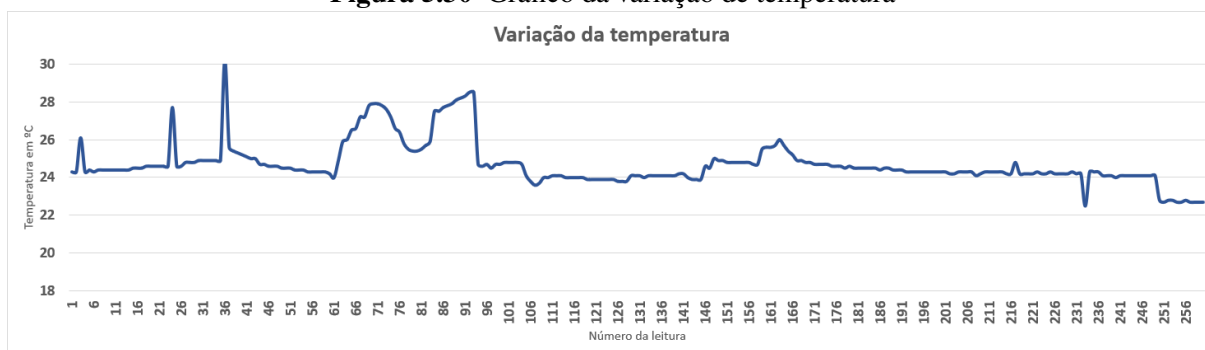
**Figura 5.29** Sistema montado em prateleira



Fonte: Autor

O gráfico abaixo (figura 5.30) mostra a variação da temperatura dentro da caixa de abelhas ao longo das leituras realizadas. Observa-se que a temperatura flutua entre 22°C e 30°C, com picos mais altos, como próximo à leitura 36, possivelmente devido à maior atividade das abelhas ou fatores externos, como luz solar. Após os picos iniciais, a temperatura estabiliza em torno de 24°C a 26°C, sugerindo condições controladas pela colmeia. No final, há uma leve queda, que pode estar associada ao resfriamento externo ou à redução da atividade interna. Essa análise destaca a importância do monitoramento para garantir um ambiente ideal às abelhas.

**Figura 5.30** Gráfico da variação de temperatura

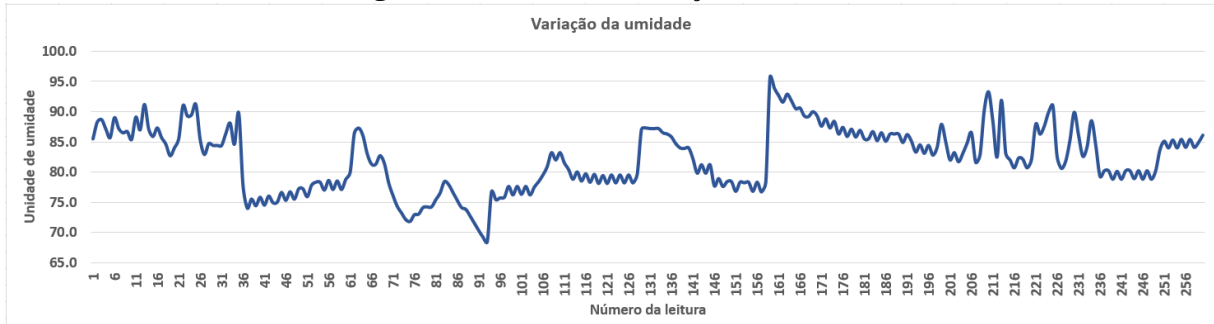


O gráfico de variação de umidade (figura 5.31) mostra um comportamento dinâmico dentro da caixa de abelhas jataí, com flutuações notáveis ao longo das leituras. Os valores oscilam entre 70 e 90 de umidade relativa, evidenciando picos e quedas significativos em determinados pontos.

Na comparação com o gráfico de temperatura anterior (figura 5.29), observa-se uma possível correlação: enquanto a temperatura apresenta certa estabilidade com declínio suave após os picos iniciais, a umidade tem oscilações mais acentuadas. Isso pode ser explicado pela relação inversa natural entre temperatura e umidade em ambientes confinados à medida que a temperatura sobe, a capacidade do ar de reter umidade aumenta, assim possivelmente resultando em quedas relativas nos níveis de umidade.

Uma possível interpretação para os picos no gráfico de umidade é a presença de fatores externos, como chuvas nos dias próximos à leitura, a umidade que podem ter influenciado o microclima dentro da caixa. Já os declínios bruscos podem indicar eventos como aumento de ventilação pelas abelhas.

**Figura 5.31** Gráfico da variação de umidade



## 6

### Conclusão

Em síntese conclui-se que, com o projeto, foi possível atingir nosso objetivo geral com sucesso. A integração entre os sensores utilizados no projeto correspondeu conforme o esperado. O foco principal está na implementação de um sistema baseado em Internet das Coisas (IoT) utilizando a plataforma Arduino, proporcionando um monitoramento remoto eficiente das condições do apiário. A junção de eletrônica, programação e práticas em apicultura trará mais avanços e melhorias a produtividade dessa atividade.

Além disso, futuras colaborações estão previstas para aprimorar ainda mais o projeto. Foi produzido um app em Flutter, chamado Appis, com diversas funções pensadas nos apicultores. Nesse app (de autoria de outro trabalho de conclusão de curso de temática similar e do IFB), será mostrado os dados dos sensores e eles estarão atrelados à uma colmeia, facilitando o acesso e a gestão dos dados pelos apicultores. Essa integração proporcionará uma ferramenta ainda mais completa e acessível, ampliando o alcance e a eficiência do sistema de monitoramento para novos horizontes e novas possibilidades, contribuindo na produtividade e saúde das abelhas. Espera-se que o tema tome mais conhecimento pelos alunos do IFB, e que esses possam participar e contribuir ao projeto.

Quanto ao circuito desenvolvido, espera-se que melhorias futuras venham a ser desenvolvidas. Seria interessante, por exemplo, que as células de carga utilizadas estivessem melhores calibradas, fornecendo valores mais consistentes. Assim como, a implementação de um outro componente eletrônico que possa vir a contribuir significativamente para o monitoramento da saúde dos apiários.

É importante reconhecer as limitações e desafios encontrados ao longo do desenvolvimento. As variáveis ambientais são vastas e complexas. Embora fatores como temperatura tenham sido abordados com sucesso, há ainda um campo extenso para aprimoramentos. A implementação prática, com a validação do sistema em condições reais, realizada em colaboração com apicultores como Alan, mostrou-se promissora. Sua validação positiva fornece uma perspectiva encorajadora, destacando a utilidade prática e a viabilidade do sistema proposto. Com a continuidade das pesquisas e o aperfeiçoamento das tecnologias, espera-se que o projeto possa contribuir significativamente para a modernização da apicultura.

## Referências

- ABREU, C. S. P. de; PEREZ, A. L. F. Redução do Consumo Energético de um Sensor Sem Fio Utilizado na Automação de um Abrigo de Cultivo. **Anais do Computer on the Beach**, [S.l.], p.220–229, 2017.
- ALMEIDA, G. d. Fatores que interferem no comportamento enxameatório de abelhas africanizadas. **Tese de Doutorado apresentada à FFCLRP-USP**, [S.l.], p.128, 2008.
- ARAÚJO, F. et al. Valoração econômica do mel no estado de Goiás: conservação e renda. **Agrarian Academy**, [S.l.], v.2, n.04, 2015.
- ARDUINO. **UNO R3 SMD**. Disponível em <https://docs.arduino.cc/hardware/uno-rev3-smd/>. Acessado 20 de maio de 2024.
- BUDEL, A. **Traité de Biologie de l’Abeille**. [S.l.: s.n.], 1968. 1–52p. v.4.
- CARVALHO, M. D. F. d. Temperatura da superfície corpórea e perda de calor por convecção em abelhas (*Apis mellifera*) em uma região semi-árida. , [S.l.], 2009.
- CIRCUITO, C. **Balança com Célula de Carga e HX711**. Disponível em <https://curtocircuito.com.br/blog/Categoria%20Arduino/balanca-com-celula-de-carga-e-hx711>. Acessado 20 de maio de 2024.
- DA ROSA, J. et al. Polinizadores em perigo: por que nossas abelhas estão desaparecendo? , [S.l.], 2017.
- DUTRA, T. F. S. **Beehiveior-sistema de monitoramento e controle de colmeias de produção apícola**. 2016. Dissertação (Mestrado em Ciência da Computação) — Brasil.
- ECHAVARRÍA, J. P. M. et al. Sistema de monitoreo de un apiario basado en el internet de las cosas. **Encuentro Internacional de Educación en Ingeniería**, [S.l.], 2022.
- ELETROGATE. **Módulo WiFi ESP32 Bluetooth 30 pinos**. Disponível em <https://www.eletrogate.com/modulo-wifi-esp32-bluetooth-30-pinos>. Acessado em 8 de agosto de 2024.
- ELLIS, M. B. et al. **Homeostasis: humidity and water relations in honeybee colonies (apis mellifera)**. 2008. Tese (Doutorado em Ciência da Computação) — University of Pretoria.
- FITZGERALD, D. W. et al. Design and development of a smart weighing scale for beehive monitoring. In: IRISH SIGNALS AND SYSTEMS CONFERENCE (ISSC), 2015. **Anais...** [S.l.: s.n.], 2015. p.1–6.
- HERAN, H. **Traité de Biologie de l’Abeille**. [S.l.: s.n.], 1968. 173–179p. v.2.
- HUMAN, H.; NICOLSON, S. W.; DIETEMANN, V. Do honeybees, *Apis mellifera scutellata*, regulate humidity in their nest? **Naturwissenschaften**, [S.l.], v.93, p.397–401, 2006.
- INCAPER. **Apicultura e Meliponicultura**. Disponível em <https://incaper.es.gov.br/apicultura>. Acessado 04 de agosto de 2023.

- JONES, J. C.; OLDROYD, B. P. Nest thermoregulation in social insects. **Advances in insect physiology**, [S.l.], v.33, p.153–191, 2006.
- LOURENÇO, M. S. M.; OLIVEIRA CABRAL, J. E. de. Apicultura e sustentabilidade: visão dos apicultores de sobral (ce). **Revista em Agronegócio e Meio Ambiente**, [S.l.], v.9, n.1, p.93–115, 2016.
- MAKERHERO. **Placa ESP32 Bluetooth**. Disponível em <https://www.makehero.com/produto/modulo-wifi-esp32-bluetooth/>. Acessado em 8 de agosto de 2024.
- MAKERHERO. **Sensor de Umidade e Temperatura DHT22**. Disponível em <https://www.makehero.com/produto/sensor-de-umidade-e-temperatura-am2302-dht22/>. Acessado 20 de maio de 2024.
- MATTERN, F.; FLOERKEMEIER, C. From the Internet of Computers to the Internet of Things. In: SACHS, K.; PETROV, I.; GUERRERO, P. (Ed.). **From Active Data Management to Event-Based Systems and More: papers in honor of alejandro buchmann on the occasion of his 60th birthday**. Berlin: Springer, 2010.
- MURPHY, F. E. et al. Big Brother for Bees (3b)—Energy Neutral Platform for Remote Monitoring of Beehive Imagery and Sound. In: ADVANCES IN SENSORS AND INTERFACES (IWASI), 2015 6TH IEEE INTERNATIONAL WORKSHOP ON. **Anais...** [S.l.: s.n.], 2015. p.106–111.
- NEXXTO. **A evolução dos sensores de monitoramento**. Disponível em <https://nexxto.com/a-evolucao-dos-sensores-de-monitoramento/>. Acessado 21 de novembro de 2023.
- POSSANI, F. A. A. et al. ABELHAS: importância ecológica e conservação. **ANAIS DO SEMEX**, [S.l.], n.12, 2019.
- POTTS, S. G. et al. Global pollinator declines: trends, impacts and drivers. **Trends in ecology & evolution**, [S.l.], v.25, n.6, p.345–353, 2010.
- SANTOS, C. S. dos; SOUZA RIBEIRO, A. de. Apicultura uma alternativa na busca do desenvolvimento sustentável. **Revista verde de agroecologia e desenvolvimento sustentável**, [S.l.], v.4, n.3, p.1, 2009.
- SILVA, A. L. **Monitoramento Não Invasivo de Colmeias Através da IoT**. 2017. Dissertação (Mestrado em Ciência da Computação) — .
- SOUZA, D. L.; EVANGELISTA-RODRIGUES, A.; CALDAS PINTO, M. d. S. de. As abelhas como agentes polinizadores. **REDVET. Revista electrónica de Veterinária**, [S.l.], v.8, n.3, p.1–7, 2007.
- TOMAZINI, C. G.; FÁTIMA GROSSI, S. de. A importância da apicultura para o agronegócio brasileiro. , [S.l.], 2019.
- VIEIRA, H. J. et al. Análise Descritiva da Interação Entre Variação de Peso de uma Colmeia, Entalpia da Atmosfera e Sistemas Meteorológicos no Sul de Santa Catarina. In: VIII ESBMET. **Anais...** [S.l.: s.n.], 2019.