

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS  
GERAIS – CAMPUS FORMIGA  
BACHARELADO EM ENGENHARIA ELÉTRICA**

**MATHEUS VAZ AMÉRICO**

**RASPBERRY PI COMO FERRAMENTA DE AUTOMAÇÃO DE CONTROLE  
LEITEIRO PARA PEQUENAS PROPRIEDADES RURAIS**

**FORMIGA – MG**

**2023**

**MATHEUS VAZ AMÉRICO**

**RASPBERRY PI COMO FERRAMENTA DE AUTOMAÇÃO DE CONTROLE  
LEITEIRO PARA PEQUENAS PROPRIEDADES RURAIS**

Trabalho de Conclusão de Curso apresentado ao  
Curso de Engenharia Elétrica do Instituto Federal  
de Minas Gerais, como requisito para obtenção do  
título de bacharel em Engenharia Elétrica.

**Orientador:** Prof. Me Marcus Vinícius de Paiva

**FORMIGA – MG**

**2023**

Américo, Matheus Vaz

A512r      Raspberry PI como ferramenta de automação de controle leiteiro para  
pequenas propriedades rurais / Matheus Vaz Américo – Formiga: IFMG, 2023.  
85p. : il. color.

Orientador: Prof. Me. Marcus Vinícius de Paiva

Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica)  
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – *Campus*  
Formiga.

1. Raspberry PI. 2. Automação. 3. Controle Leiteiro. 4. Célula de Carga. 5.  
RFID. I. Paiva, Marcus Vinícius de. II. Título.

CDD 621.3

*Ficha catalográfica elaborada pela Bibliotecária Lívia Renata Santos- CRB/6-2561*

MATHEUS VAZ AMÉRICO

RASPBERRY PI COMO FERRAMENTA DE AUTOMAÇÃO DE CONTROLE  
LEITEIRO PARA PEQUENAS PROPRIEDADES RURAIS

Trabalho de Conclusão de Curso  
apresentado ao Curso de Engenharia  
Elétrica do Instituto Federal de Minas  
Gerais como requisito para obtenção do  
Título de Bacharel em Engenharia  
Elétrica.

Avaliado em: 30 de novembro de 2023.

Nota: 83

**BANCA EXAMINADORA**

*Marcus Vinicius de Paiva*

Prof. Me. Marcus Vinicius de Paiva (Orientador)

*José Antônio Moreira de Rezende*

Prof. Me. José Antônio Moreira de Rezende

*Andreza Patrícia Batista*

Eng. Andreza Patrícia Batista

## RESUMO

Este projeto visa empregar o *Raspberry Pi* como uma ferramenta de automação para o controle leiteiro em pequenas propriedades rurais. A proposta é utilizar a versatilidade e eficiência da plataforma para desenvolver um sistema de monitoramento abrangente, proporcionando aos produtores rurais uma maneira automatizada de acompanhar a produção de leite em suas propriedades. A implementação envolverá a integração de sensores de temperatura e umidade ao *Raspberry Pi*, responsáveis por coletar dados cruciais relacionados à produção leiteira. Esses sensores registrarão informações como a quantidade de leite produzido e características que podem alterar a produção, possibilitando uma análise detalhada do desempenho do rebanho. O sistema resultante terá a capacidade de fornecer informações de uma maneira simples e clara através de um site *web*, capacitando os produtores a tomar decisões informadas sobre o manejo e a otimização da produção leiteira. Além disso, a facilidade de integração do equipamento com outros dispositivos permitirá uma expansão futura do sistema, enquanto a acessibilidade remota aos dados oferecerá conveniência aos produtores, tornando essa solução uma opção viável e acessível para melhorar a eficiência nas pequenas propriedades rurais.

**Palavras chave:** *Raspberry Pi*. Automação. Controle leiteiro.

## **ABSTRACT**

This project aims to employ the Raspberry Pi as an automation tool for dairy management in small rural properties. The proposal is to harness the versatility and efficiency of the platform to develop a comprehensive monitoring system, providing farmers with an automated way to track milk production on their properties. The implementation will involve the integration of temperature and humidity sensors with the Raspberry Pi, responsible for collecting crucial data related to dairy production. These sensors will record information such as the quantity of milk produced and characteristics that can impact production, enabling a detailed analysis of herd performance. The resulting system will have the capability to provide information in a simple and clear manner through a website, empowering farmers to make informed decisions regarding the management and optimization of milk production. Furthermore, the ease of integrating the equipment with other devices will allow for future system expansion, while remote accessibility to data will offer convenience to farmers, making this solution a viable and accessible option to enhance efficiency in small rural properties.

**Keywords:** Raspberry Pi, Automation, Dairy Control.

## LISTA DE FIGURAS

Figura 1 - Medidor de nível do tipo flutuador .....	19
Figura 2 - Medidor de nível por pressão hidrostática .....	20
Figura 3 - Medidor de nível por capacitância .....	21
Figura 4 - Medidor de nível por pesagem .....	23
Figura 5- Circuito básico da ponte de <i>Wheastone</i> .....	24
Figura 6 - Exemplos de precisão e exatidão .....	25
Figura 7- Modelo básico do funcionamento de um sistema RFID .....	27
Figura 8 - Sistema leitor-tag do modelo <i>near-field</i> .....	28
Figura 9 - Sistema leitor-tag do modelo <i>far-field</i> .....	29
Figura 10 - Formas de energia em um sensor .....	29
Figura 11 - Variação de uma grandeza física em um sensor analógico .....	30
Figura 12- Leitura da posição de um objeto por um <i>encoder</i> incremental .....	31
Figura 13 - Estrutura de programa típico em Python .....	32
Figura 14 - Componentes de um sistema de banco de dados .....	35
Figura 15 - DER (Diagrama Entidade- Relacionamento) .....	36
Figura 16 - Exemplos de tabelas em SGBD relacional .....	37
Figura 17 - Fluxograma de desenvolvimento do protótipo .....	39
Figura 18 - <i>Raspberry Pi 3 Model B+</i> .....	41
Figura 19 – Identificação dos pinos GPIO do <i>Raspberry Pi</i> .....	42
Figura 20 - Módulo leitor RFID MFRC522 .....	43
Figura 21 - <i>Tag</i> chaveiro RFID .....	44
Figura 22 - Esquema de conexão para o módulo MRFC522 .....	45
Figura 23 - Comando para instalação das ferramentas necessárias .....	45
Figura 24 - Comando para instalar bibliotecas necessárias .....	45
Figura 25 - DHT11 .....	47
Figura 26 - Esquema de conexão para o DHT11 .....	48
Figura 27 - Célula de carga de 20 kg .....	49
Figura 28 - HX711 .....	50
Figura 29 - Balança desenvolvida .....	51
Figura 30 - Ligação elétrica entre os objetos da balança .....	52
Figura 31 – Gráfico de calibração da balança .....	53
Figura 32 - Tela inicial do apache apresentada pelo VNC- <i>viewer</i> após instalação .....	54

Figura 33- Estrutura da tabela criada.....	55
Figura 34 - Tabela Criada no <i>Phpmyadmin</i> .....	55
Figura 35 - Comandos para obter/ parar os programas <i>hostapd</i> e <i>dnsmasq</i> . ....	56
Figura 36 - Comandos para configurar um IP estático para a interface <i>wlan0</i> .....	56
Figura 37 - Configuração do servidor DHCP .....	57
Figura 38 - Configurações do ponto de acesso.....	58
Figura 39 – Comandos para organizar a tabela de IP's .....	58
Figura 40 - Página de detalhes do projeto .....	59
Figura 41 - Página de dados do desenvolvedor .....	60
Figura 42 - Página de dados do animal Esmeralda disposto em tabela.....	61
Figura 43 – Página dedicada a apresentação das informações em forma de gráficos .....	62
Figura 44 - Esquema elétrico do projeto .....	63
Figura 45 - Protótipo construído.....	64
Figura 46 - Protótipo em funcionamento no momento da ordenha.....	68
Figura 47 - Gráfico de produção da vaca Nambuca .....	71
Figura 48 - Gráfico de produção total no período de 5 dias .....	72
Figura 49 - Dados da produção da vaca Nambuca dispostos na forma tabular.....	73
Figura 50 - Dados da produção total dispostos em tabela .....	74

## LISTA DE TABELAS

Tabela 1 - Equipamentos de medição direta/indireta. ....	18
Tabela 2 - Seleção de instrumentos de nível .....	22
Tabela 3 - Especificações técnicas do <i>Raspberry Pi 3 B+</i> . ....	40
Tabela 4 - Especificações técnicas módulo leitor <i>RFID MFRC522 Mifare</i> .....	43
Tabela 5 - Especificações técnicas DHT11 .....	47
Tabela 6 - Especificações técnicas célula de carga 20 kg. ....	49
Tabela 7 - Medições dos 3 pesos de prova na balança desenvolvida .....	65
Tabela 8 – Resumo dos valores para determinar a exatidão.....	66
Tabela 9- Relação <i>tag</i> e nome dos animais para a Fazenda .....	67
Tabela 10 - Dados obtidos relacionados com as <i>tags</i> para a Fazenda.....	70
Tabela 11 - Levantamento de custos para o protótipo.....	75

## LISTA DE ABREVIATURAS E SIGLAS

ANEEL – Agência Nacional de Energia Elétrica

ARM – Acorn RISC Machine

BD – Banco de Dados

BLE – Bluetooth Low Energy

CSS – Cascading Style Sheets

CSI – Camera Serial Interface

DC – Direct Current

DHCP – Dynamic Host Configuration Protocol

DSI – Display Serial Interface

DNS – Domain Name System

EMBRAPA – Empresa Brasileira de Pesquisa Agropecuária

ER – Entidade-Relacionamento

EPC – Electronic Product Code

GPIO – General Purpose Input/Output

HTML – HyperText Markup Language

IBGE – Instituto Brasileiro de Geografia e Estatística

IP – Internet Protocol

LAMP – Linux, Apache, MySQL, PHP

MAC – Media Access Control

MISO – Master In Slave Out

MOSI – Master Out Slave In

NOOBS – New Out Of the Box Software

NXP – Next Experience Point

RFID – Radio-Frequency Identification

SCK – Serial Clock

SDA – Serial Data Line

SGBD – Sistema de Gerenciamento de Banco de Dados

RFID – Radio Frequency Identification

PHP – Hypertext Preprocessor

URL – Uniform Resource Locator

VCC – Voltage at the Common Collector

VNC – Virtual Network Computing

GND – Ground

URL – Uniform Resource Locator

SSID – Service Set Identifier

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>14</b>
<b>1.1 Justificativa .....</b>	<b>15</b>
<b>1.2 Objetivos gerais .....</b>	<b>16</b>
<b>1.3 Objetivos específicos.....</b>	<b>16</b>
<b>1.4 Estrutura do Trabalho .....</b>	<b>17</b>
<b>2 REFERENCIAL TEÓRICO .....</b>	<b>18</b>
<b>2.1 Medições de nível.....</b>	<b>18</b>
<b>2.1.1 Dispositivos do tipo pesagem .....</b>	<b>22</b>
<b>2.2 Precisão, exatidão e terminologia nas medições .....</b>	<b>25</b>
<b>2.3 Tecnologia RFID.....</b>	<b>26</b>
<b>2.3.1 Componentes do RFID .....</b>	<b>27</b>
<b>2.3.2 Tipos de RFID .....</b>	<b>27</b>
<b>2.4 Sensores .....</b>	<b>29</b>
<b>2.4.1 Sensores analógicos.....</b>	<b>30</b>
<b>2.4.2 Sensores digitais .....</b>	<b>30</b>
<b>2.5 Linguagem Python .....</b>	<b>31</b>
<b>2.6 Sites em redes.....</b>	<b>32</b>
<b>2.6.1 PHP .....</b>	<b>33</b>
<b>2.6.2 HTML.....</b>	<b>33</b>
<b>2.6.3 CSS.....</b>	<b>34</b>
<b>2.7 Banco de dados.....</b>	<b>35</b>
<b>3 MATERIAIS E MÉTODOS .....</b>	<b>38</b>
<b>3.1 Raspberry Pi.....</b>	<b>40</b>
<b>3.2 Módulo MFRC522.....</b>	<b>43</b>
<b>3.2 DHT11.....</b>	<b>46</b>

3.3 Balança com células de carga e HX711 .....	48
3.4 Criação do LAMP .....	53
3.5 Raspberry PI como ponto de acesso .....	55
3.6 Desenvolvimento do <i>website</i> .....	59
3.7 Conclusão do protótipo .....	63
3.8 Características da balança .....	65
3.9 Coleta de dados .....	67
4 RESULTADOS E DISCUSSÕES .....	69
4.1 Levantamento de custos .....	75
5 CONCLUSÃO.....	76
6 TRABALHOS FUTUROS .....	78
REFERÊNCIAS BIBLIOGRÁFICAS .....	79
ANEXO A.....	82
ANEXO B.....	83
ANEXO C.....	84

## 1 INTRODUÇÃO

O Brasil foi o quinto maior produtor de leite bovino do mundo, produzindo 35,3 bilhões de litros em 2021. Essa produção representou uma queda de 0,03 % em relação ao ano anterior e apresenta queda após a alta de 1,2 % ocorrida em 2019. Adicionalmente, as regiões Sul e Sudeste ocupam a liderança sendo responsáveis por 67,8 % da produção nacional, sendo o estado de Minas Gerais responsável por 9,6 bilhões de litros em 2021, correspondendo a 27,19% da produção nacional (IBGE, 2023).

Apesar do Brasil estar nas primeiras posições do *ranking* de produção de leite mundial, no quesito de produtividade animal, o país ocupa a posição 84 no mundo, com uma produtividade de 2.214 litros/animal/ano. Os Estados Unidos ultrapassam uma produção de 10.000 litros/animal/ano (EMBRAPA, 2020).

Foi observado um aumento na produção nacional de leite nos últimos anos, devido a implantação constante e diversificada de recursos tecnológicos nas fazendas (EMBRAPA, 2022). A pecuária de precisão, neste caso, é uma abordagem utilizada com o intuito de maximizar os indicadores de desempenho de produção. Ela permite tomadas de decisão baseadas em dados, visando adotar tecnologias acessíveis e voltadas para um melhor gerenciamento do rebanho. Essas tecnologias promovem melhorias na eficiência da gestão e também aumentos expressivos em volume e produtividade leiteira. Geralmente, a pecuária de precisão conta com sensores instalados nas fazendas, que coletam dados e, após processados, se transformam em informações úteis para a tomada de decisões (EMBRAPA, 2022).

Em propriedades com alta produtividade, o uso de tecnologias no manejo é indispensável, sendo que, tanto a produção quanto a qualidade são monitoradas diariamente para melhorar o manejo do gado.

Em algumas propriedades existem diversas técnicas para o conforto e bem estar dos animais, onde eles podem receber camas individuais e galpões com sistema de ventilação cruzada para controlar a temperatura do ambiente. Um exemplo é a fazenda Colorado localizada em Araras, Estado de São Paulo, que possui produção diária de 67.640 litros/dia e as novilhas produzem em média de 45 quilos de leite por dia. Uma característica é a utilização de bicos de água quando a temperatura excede 24 °C e, também, um sistema de ventilação cruzada para manter a temperatura estável (MILKPOINT, 2023).

O controle leiteiro é uma prática de aferição da produção de leite de uma vaca em fase de lactação. Este fornece dados técnicos que permitem tomada de decisão tanto para um

melhoramento animal quanto para melhora na qualidade da nutrição para as vacas em lactação. Entre as vantagens de adoção desse método estão (i) diagnóstico, planejamento e melhorias no processo produtivo, (ii) Estabelecimento de critério para descarte de matrizes e, principalmente, (iii) definir a quantidade de ração de acordo com a produção de cada vaca (EMBRAPA, 2016).

Por outro lado, pequenos produtores com produção de até 500 litros/dia geralmente não são capazes de realizar esse tipo de controle de produção leiteira de maneira a não atrasar o período de ordenha e equipamentos para auxiliar nesse processo geralmente não são atrativos do ponto de vista financeiro para esse nicho de produtores.

A fim de proporcionar agilidade nessa técnica de controle do processo de produção, neste Trabalho de Conclusão de Curso é apresentada uma solução técnica empregando *Raspberry Pi* com o intuito de: (i) obtenção dos dados de pesagem das latas e (ii) obtenção de dados de temperatura e umidade. Desta forma, será possível criar uma base de dados com as informações de produção de leite de cada vaca do rebanho juntos a temperatura e umidade no momento da ordenha, facilitando os processos de decisão pertinentes a correta interpretação dessas informações.

O *Raspberry Pi*, um microcomputador de baixo custo e alto desempenho, tem se destacado como uma ferramenta versátil para a criação de sistemas eletrônicos e projetos de automação. Uma aplicação significativa dessa tecnologia é encontrada na integração com balanças equipadas com células de carga. As células de carga, sensores de peso sensíveis e precisos, desempenham um papel fundamental em dispositivos de medição de peso. Ao incorporar o *Raspberry Pi* nesses sistemas, é possível criar soluções inteligentes que oferecem funcionalidades avançadas, como monitoramento remoto, armazenamento de dados em nuvem e integração com outros dispositivos, ampliando significativamente as capacidades e a eficiência dessas balanças.

## **1.1 Justificativa**

A falta de acesso a informação e novas tecnologias são alguns dos fatores que ocasionam a baixa produtividade de pequenos produtores. Estes não dispõem de ferramentas efetivas para o controle leiteiro, visto que equipamentos que permitem a pesagem de leite e monitoramento de dados externos são caros e não atrativos para o pequeno produtor, e quando o controle leiteiro

é realizado manualmente ocasiona atrasos no período de ordenha e por isso é realizado em intervalos maiores e tempo ou muitas vezes nem é realizado.

Portanto, quando essa técnica não é realizada ou é feita de maneira incorreta ocasiona dificuldades para melhorar a genética do rebanho, e controlar a quantidade de concentrado de cada animal.

Entretando, ela não é muito utilizada entre os pequenos produtores. Utilizar um equipamento capaz de simplificar a execução dessa técnica torna possível ter um controle mais refinado da produção, e por consequência tomar as devidas decisões de manejo com o intuito de aumentar a produtividade. Tais como, substituir animais em lactação com baixo rendimento por animais de melhor rendimento, ou realizar uma melhor divisão de concentrado entre os animais.

Fatores externos como temperatura e umidade também podem ser indicadores de queda ou aumento da produção, e, portanto, também serão monitorados.

## 1.2 Objetivos gerais

Visto que a baixa produtividade está relacionada a falta de acesso as novas tecnologias, este Trabalho de Conclusão de Curso visa desenvolver um protótipo capaz de contribuir com o processo de controle leiteiro, fornecendo de forma precisa, a produção diária de leite bovino, além de dados como temperatura e umidade. Esses valores serão armazenados em memória física e dispostos em rede local gerenciada pelo equipamento. A correta análise desses dados permite que profissionais da área zootécnica conduzam técnicas de manejo e rearranjo com o intuito de aumentar a produção de leite.

## 1.3 Objetivos específicos

Os objetivos específicos são listados a seguir:

- Desenvolver a estrutura física da balança;
- Acoplar uma célula de carga à balança;
- Programar o dispositivo *Raspberry Pi* para aquisição e armazenagem dos dados de pesagem do leite advindos da célula de carga, dos dados ambientais de temperatura e umidade do local da ordenha e os associando aos animais por meio da tecnologia de *Radio Frequency Identification* (RFID);

- Disponibilizar página web para apresentação dos dados de produção e das informações ambientais.

#### **1.4 Estrutura do Trabalho**

Este trabalho está dividido em 6 capítulos. No capítulo 1 é apresentado um contexto envolvendo o cenário da produção de leite no Brasil, bem como a importância da automatização de processos no setor, visando o aumento da confiabilidade da produção. O capítulo 2 apresenta o referencial teórico que descreve todas as ferramentas utilizadas no desenvolvimento da proposta. No capítulo 3 são apresentados os equipamentos utilizados e as etapas principais de sua utilização, no protótipo do sistema de automação do controle leiteiro. O capítulo 4 é destinado a apresentação dos resultados e as discussões acerca da eficácia da proposta. Por fim, no capítulo 5, são apresentadas as conclusões sobre todos os levantamentos e observações após os estudos e testes realizados, bem como é apresentado, no capítulo 6, propostas de trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Este capítulo trata das principais ferramentas que foram utilizadas para o desenvolvimento do projeto. Nele serão apresentadas de forma sucinta: as medições de nível com foco principal em células de carga; microcomputador (*Raspberry PI*); sensor RFID; sensor de umidade e temperatura; *websites*; redes locais; a linguagem de programação *Python* e as estruturas de banco de dados.

### 2.1 Medições de nível

A variável nível é comumente utilizada em aplicações industriais, ela pode ser determinada como a posição de uma interface entre dois meios distintos (BEGA, 2011), ou simplesmente a altura do conteúdo de um reservatório este podendo ser líquido ou sólido (SENAI, 1999).

A seleção do equipamento para medição de nível deve partir das características de cada prática, tais como o produto que se deseja obter o nível, a exatidão e também custos e restrições de cada aplicação (BEGA, 2011). Existem dois métodos de medição de nível, o método de medição direta onde o nível é definido unicamente pela posição do plano superior do produto medido e o método de medição indireta onde nível é determinado de acordo com uma segunda variável (empuxo, capacitância ou peso, por exemplo). A Tabela 1 mostra alguns métodos de medição de nível e suas classificações (SENAI, 1999).

Tabela 1 - Equipamentos de medição direta/indireta.

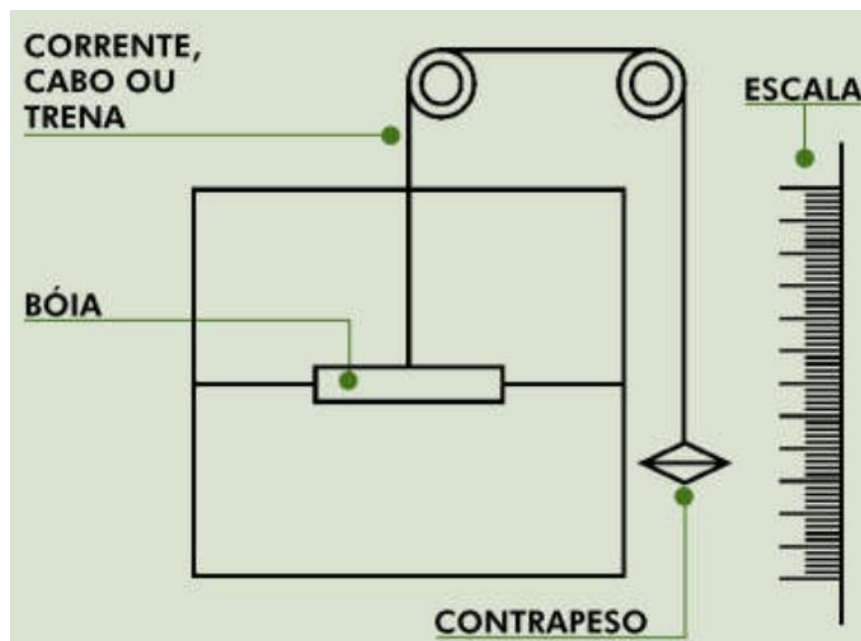
MEDIÇÃO DIRETA	MEDIÇÃO INDIRETA
	Deslocador (empuxo)
	Pressão diferencial (diafragma)
Réguas ou Gabaritos	Capacitância eletrostática
Visores de Nível	Radar
Boia ou Flutuador	Ultrassônico
	Pás rotativas
	Por pesagem (célula de carga)

Fonte: Senai (2019).

Réguas, gabaritos e visores de nível são equipamentos que tem como finalidade o monitoramento do nível de líquidos ou de alguma interface. Uma interface ocorre quando em um mesmo reservatório ocorre a mistura de dois fluidos de densidades diferentes, sendo alguns exemplos a interface óleo-água, óleo-gás e líquido-sólido. Tais equipamentos, geralmente, são de baixo custo e são utilizados amplamente para monitoramento local, exceto em locais onde pressão ou temperatura impeçam sua utilização, por exemplo, em uma caldeira onde a temperatura é elevada. Neste caso, no visor de nível, a temperatura do fluido dentro do visor é diferente da temperatura no interior da caldeira, o que altera a densidade dos fluidos nesses locais, gerando uma incorreta medição de nível (BEGA, 2011).

As boias ou flutuadores são equipamentos bem simples que são fixados a um cabo o qual, por sua vez, possui a extremidade ligada a um contrapeso. Nesse contrapeso é fixado um ponteiro que indica o nível em uma escala (JUNIOR, 2017). A Figura 1 apresenta um exemplo genérico desse tipo de flutuador.

Figura 1 - Medidor de nível do tipo flutuador



Fonte: Junior (2021).

Em equipamentos do tipo pressão diferencial, o nível é determinado indiretamente pelo valor da pressão exercida pela coluna do líquido. A relação entre a pressão e a altura da coluna independe da largura ou formato do reservatório. Para esse tipo de medição é utilizada a Equação 1 para determinar o nível. A Figura 2 ilustra um modelo genérico para esse tipo de medidor (JUNIOR, 2017):

$$P = \delta \cdot g \cdot h + P_{atm} \quad (1)$$

Sendo que:

P: Pressão total (pascal - Pa).

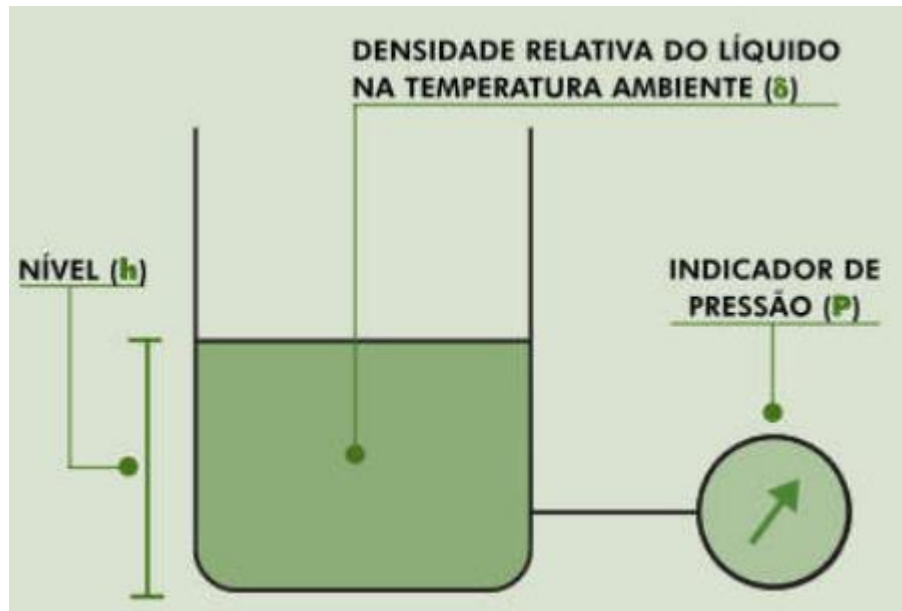
$\delta$ : Densidade do fluido (quilograma por metro cúbico - kg/m<sup>3</sup>).

g: Aceleração devido à gravidade (metros por segundo ao quadrado - m/s<sup>2</sup>).

h: Altura do fluido acima do ponto de referência (metros - m).

P atm: Pressão atmosférica (ou pressão no ponto de referência) (pascal - Pa).

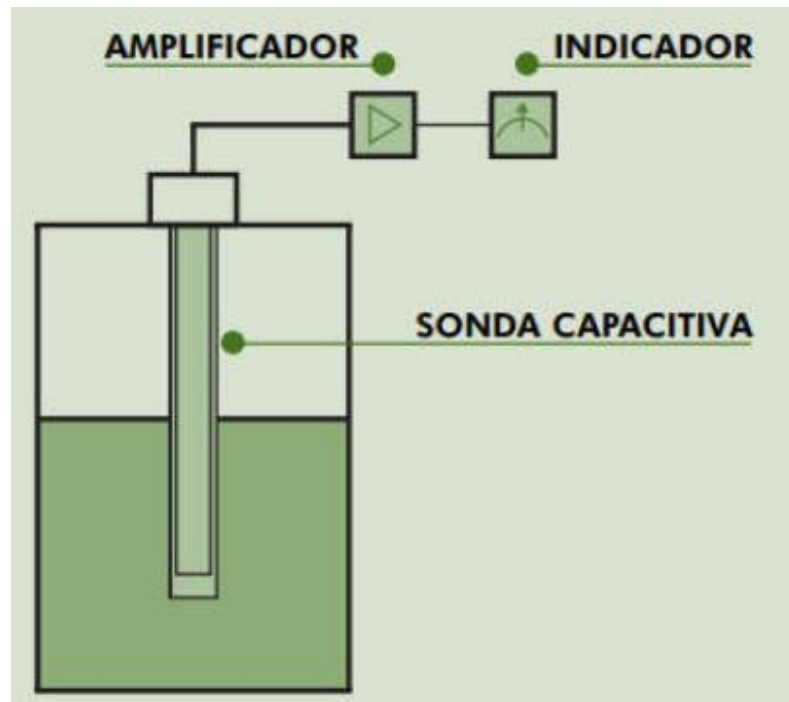
Figura 2 - Medidor de nível por pressão hidrostática



Fonte: Junior (2021)

No medidor do tipo capacitivo o nível é determinado de acordo com a capacitância entre o eletrodo submerso e as paredes do tanque. À medida que o nível do tanque aumenta, o mesmo ocorre com sua capacitância pois o dielétrico do ar é substituído pelo dielétrico do líquido (JUNIOR, 2017). A Figura 3 apresenta um modelo com sonda capacitiva:

Figura 3 - Medidor de nível por capacitância



Fonte: Junior (2021).

Os dispositivos do tipo ultrassônico, são baseados na reflexão de ondas sonoras, quando esta encontra a interface com o produto que se deseja medir o nível. Ou seja, pelo tempo necessário desde que a onda é gerada até o momento que é refletida e captada novamente. Como a velocidade de propagação da onda em um meio é constante, o tempo é proporcional entre a distância entre o sensor responsável pela captação da onda e a interface (BEGA, 2011). Logo:

$$d = v \cdot \Delta t \quad (2)$$

Sendo que:

$d$  = distância (metros - m);

$v$  = velocidade de propagação da onda (metros por segundo - m/s);

$\Delta t$  = variação de tempo (segundos - s).

Esses são alguns dos principais sensores de nível presentes no mercado e dentro de cada um deles existem diversas categorias e modelos. Para determinar qual o sensor mais adequado para cada aplicação, é preciso atentar às limitações de cada aplicação, as condições de pressão e temperatura de operação e também o produto pelo qual se deseja medir o nível. Com o intuito de realizar a devida seleção do instrumento de medição de nível, Bega (2021) disponibiliza um

guia, reproduzido na Tabela 2. Este associa o instrumento de medição de nível adequado para o tipo de aplicação.

Tabela 2 - Seleção de instrumentos de nível

Item	Tipo de Instrumento	Medição contínua				
		Transmissor / Controlador				
		1	2	3	4	5
1	Flutuador	B	-	R	R	-
2	Deslocador	E	-	B	-	-
3	Pressão diferencial	E	R	R	-	-
4	Ultrassônico	B	R	-	R	B
5	Radar	E	B	-	B	R
6	Capacitivo	B	B	-	-	B
7	Eletromecânico	B	R	B	B	B
8	Pás rotativas	-	-	-	-	-
9	Pesagem	B	-	R	R	B

Nota: B = bom; E =excelente; R =ruim, E – sem muitas aplicações práticas.

1 = líquidos limpos; 2 = líquidos com espuma; 3 =interface; 4 = polpas; 5 = sólidos;

Fonte: Bega (2011).

### 2.1.1 Dispositivos do tipo pesagem

O nível de qualquer tanque pode ser determinado pela sua pesagem, subtraindo a pesagem total do tanque pelo peso do tanque completamente vazio. O elemento de medição para dispositivos do tipo pesagem são as células de carga, que são construídas a base de dispositivos do tipo *strain-gage* e projetadas a forma de produzirem um sinal elétrico proporcional à carga depositada sobre elas (BEGA, 2011). A Figura 4 ilustra um sistema de pesagem em um silo, onde a célula de carga fica na base do mesmo.

Figura 4 - Medidor de nível por pesagem

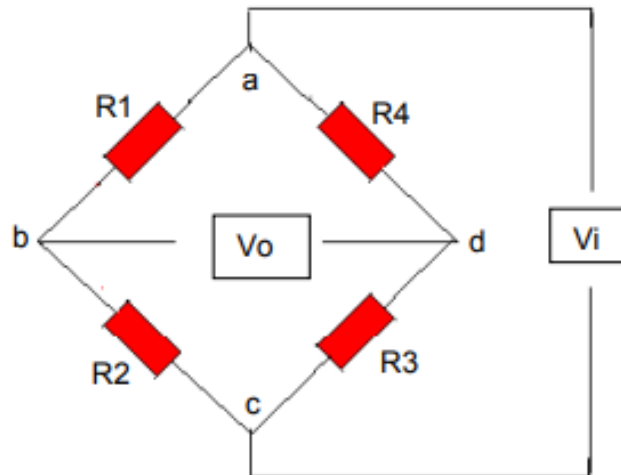


Fonte: UFRN (2013).

Os *Strain-gages* são equipamentos resistivos analógicos, que são fixados na estrutura da célula de carga afim de variar sua resistência de acordo com a força de tensão ou compressão que a célula recebe. A variação da sua resistência ocorre de maneira proporcional a deformação do corpo da célula de carga (PORTAL CELULA DE CARGA, 2023).

O *strain-gage* mais utilizado é o de componentes metálicos, ele é composto por um fio muito fino ou por folhas metálicas dispostas em forma de uma grade. Nesse padrão a deformação paralela maximiza a extensão dos fios ou folhas, aumentando a sua resistência. Portanto a deformação sofrida pelo corpo da célula é transferida para o *strain-gage* (PORTAL CELULA DE CARGA, 2023). Células de carga são transdutores que convertem força ou massa em uma saída elétrica mensurável, sendo que o circuito em ponte de *Wheatstone* permite medir mudanças na resistência que ocorrem nos *strain-gages* (WEBER, 2008).

Figura 5- Circuito básico da ponte de *Wheastone*



Fonte: Weber (2008).

A Figura 5 apresenta um circuito básico de uma célula de carga onde  $R1$ ,  $R2$ ,  $R3$ ,  $R4$  são resistências elétricas, e a relação entre elas é escrita conforme a Equação (3). Essa ponte é energizada nos terminais “a” e “c” e o sinal de saída  $V_o$  é obtido nos terminais “b” e “d” (WEBER, 2008).

$$V_o = \frac{R1 \cdot R3 - R2 \cdot R4}{(R1 + R2)(R4 + R3)} \cdot V_i \quad (3)$$

Sendo que:

$V_o$  = Sinal de saída (Volts - V);

$V_i$  = Tensão de excitação (Volts - V);

$R1$ ,  $R2$ ,  $R3$ ,  $R4$  = Resistências elétricas (Ohms -  $\Omega$ ).

A forma geométrica, o material, a intensidade da força a ser medida e também as condições ambientais são os principais critérios utilizados para a escolha da célula de carga (PORTAL CELULA DE CARGA, 2023). A precisão está diretamente relacionada a quantidade de células de carga utilizadas e a sua disposição deverá ser realizada em forma geométrica regular (triângulo, quadrado, pentágono...), interligadas em série a um amplificador. A escolha

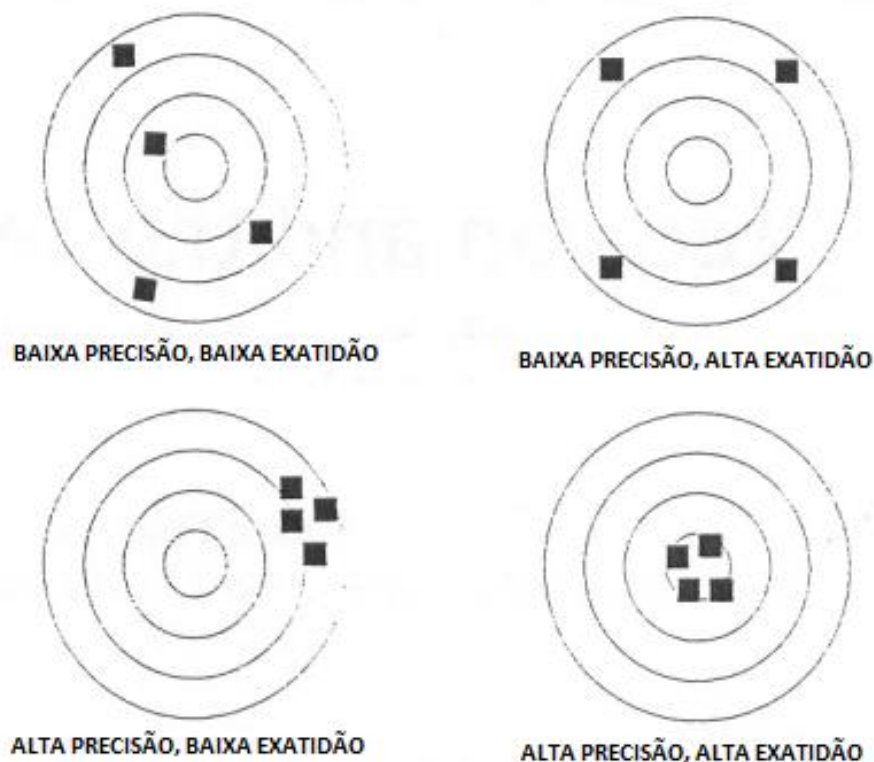
da célula de carga ideal depende da precisão necessária e de sua viabilidade econômica (BEGA, 2011).

## 2.2 Precisão, exatidão e terminologia nas medições

A precisão é uma medida da concordância entre determinações repetidas de uma mesma grandeza. Ela é usualmente quantificada como o desvio padrão de uma série de medidas. A exatidão de uma medida (ou da média de um conjunto de medidas) é a distância estimada entre a medida e um valor “verdadeiro”, “nominal”, “tomado como referência”, ou “aceito” (SILVA, 2006).

Enquanto que a precisão refere-se a concordância das medidas em uma série de medições, a exatidão visa compreender a variação entre uma série de medidas e o seu valor nominal que pode ser determinado por outro medidor. Esta primeira pode ser quantificada como o desvio padrão de uma série de medidas e a segunda como o desvio percentual de um valor já conhecido (SILVA, 2006).

Figura 6 - Exemplos de precisão e exatidão



Nos exemplos da Figura 6 podemos identificar: a) Baixa precisão e baixa exatidão – b) Baixa precisão e alta exatidão, c) Alta precisão e baixa exatidão e d) Alta precisão e alta exatidão.

Outras terminologias frequentemente utilizadas são a resolução e a sensibilidade, a resolução determina a capacidade de leitura, por exemplo em uma régua graduada em milímetros possui resolução de 0,5 mm ou menos, afinal é onde se torna possível observar a sua capacidade de medida. Já a sensibilidade é um termo utilizado para definir o menor valor mensurável por um determinado instrumento, podemos como exemplo uma balança com sensibilidade 1g, ela seria incapaz de aferir valores como 15 mg, ou 2 mg.

### 2.3 Tecnologia RFID

O termo RFID é uma abreviatura para *Radio Frequency Identification*, em português, (identificação por radiofrequência). Sua origem deu-se na segunda guerra mundial, nesta época, tinha o objeto de identificar aviões inimigos ou amigos (JUNIOR, 2007).

A partir da década de 80 essa tecnologia começou a ter novas atribuições, dentre elas a utilização em sistemas de rastreamento e controle, gerando diversos estudos e pesquisas visando desenvolver uma arquitetura capaz de utilizar todos os recursos dessa tecnologia baseada em radiofrequência. Em um desses estudos foi desenvolvido uma arquitetura de identificação de produtos que utilizava os recursos proporcionais pelos sinais de radiofrequência, este ficou conhecido como código eletrônico de produtos EPC (*Electronic Product Code*) (JUNIOR, 2007).

Basicamente o RFID é um sistema sem fio que utiliza ondas de rádio para definir a identidade de etiquetas eletrônicas (*Tags*). Essas etiquetas conseguem armazenar muitas informações, essas podendo fornecer identificação, ou localização de informações além de informações específicas sobre cada objeto (JUNIOR, 2007)

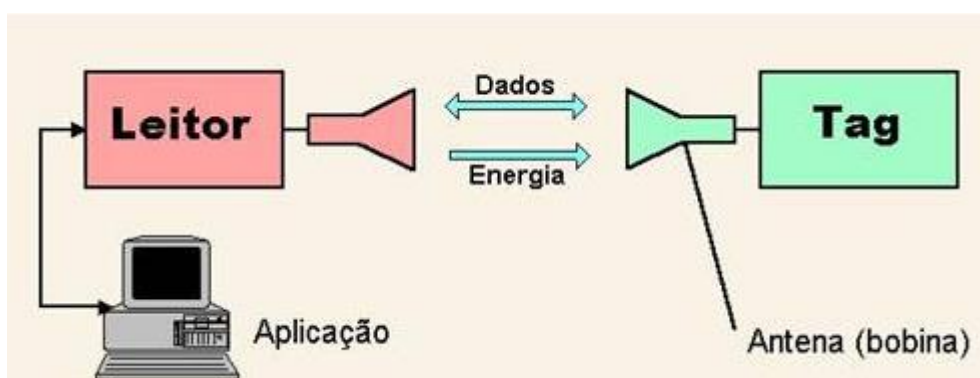
A leitura das informações presentes na *tag*, é realizada por um leitor (*reader*), e toda a comunicação ocorre através das ondas de radiofrequência, e por isso pode ocorrer em ambientes onde não é necessário o contato físico ou visual entre os aparelhos (PREDIGER, SILVEIRA e FREITAS, 2014).

### 2.3.1 Componentes do RFID

O sistema RFID é, de maneira geral, composto por dois componentes: o *transponder* (*tag*) que estará no objeto que necessita de identificação, e o leitor (*reader*), que pode ser um dispositivo de captura de dados ou de captura/transmissão de dados.

Então o *transponder* é o dispositivo que armazena os dados do sistema RFID. Geralmente é constituído de uma antena e um microchip eletrônico responsável pela identificação única do *transponder*. Existem duas categorias de RF *tags*, sendo as ativas, as *tags* alimentadas por bateria interna e permite processo de escrita e leitura, e as passivas, que não possuem fonte de energia, somente opera quando alimentada pela energia fornecida pelo leitor através das ondas eletromagnéticas (UFRJ, 2023). A Figura 7 ilustra o funcionamento dessa tecnologia.

Figura 7- Modelo básico do funcionamento de um sistema RFID



Fonte: UFRJ (2023).

### 2.3.2 Tipos de RFID

A tecnologia RFID, dada sua notável flexibilidade, é adaptável a diversas aplicações. Duas categorias essenciais para os sistemas RFID serão delineadas com base na modalidade de recebimento de energia e na extensão de sua capacidade de comunicação (LOPES, PESTRE e TORRES, 2018).

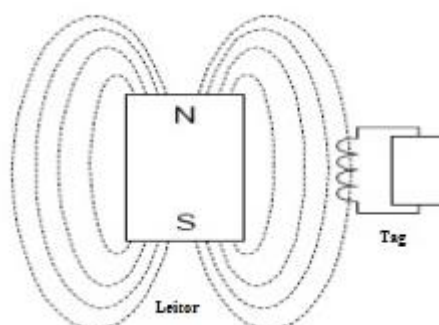
No quesito de alimentação de energia, a divisão mais importante se encontra entre os modelos ativos ou passivos. Basicamente nos modelos passivos, as *tags* não apresentam nenhuma conexão com alguma fonte de energia e nos modelos ativos, a *tag* é conectada a

alguma fonte de energia, seja ela uma bateria ou alguma fonte individual (LOPES, PESTRE e TORRES , 2018).

No modelo passivo, por não existir uma fonte de energia associada à *tag*, é necessário que o leitor alimente de alguma forma a *tag*, afim de ocorrer a transmissão de dados. Nesse quesito, existem duas maneiras principais para transferência de energia para esses sistemas, e a escolha deve-se principalmente pelo alcance desejado para a aplicação. Os dois modelos são o *near-field* e o *far-field* (LOPES, PESTRE e TORRES , 2018).

O princípio de funcionamento da tecnologia *near-field* é o de indução magnética. O leitor emite corrente alterada, resultado numa variação do campo magnético próximo ao leitor. Na *tag* existem um indutor e um capacitor, e esse indutor responde com a variação do campo magnético carregando o capacitor que quando totalmente carregado alimenta a *tag*. A Figura 8 ilustra essa tecnologia (LOPES, PESTRE e TORRES , 2018):

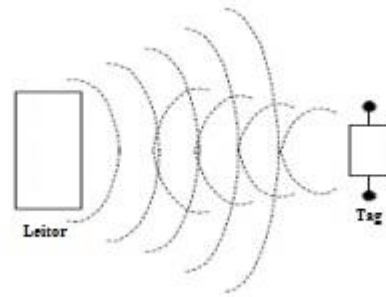
Figura 8 - Sistema leitor-tag do modelo *near-field*



Fonte: Lopes et al (2018)

A tecnologia *far-field* adota o princípio semelhante ao da *near-field* no entanto, neste modelo, são empregadas antenas dipolo tanto na *tag* quanto no leitor. Essas antenas facilitam a variação do campo magnético, resultando em uma distância de comunicação mais ampla. No referido modelo, ambas as antenas são sintonizadas na mesma frequência para otimizar a absorção da maior parte da energia presente nessa faixa de frequência. Uma representação visual desse modelo *far-field* é apresentada na Figura 9 (LOPES et al, 2018).

Figura 9 - Sistema leitor-tag do modelo *far-field*



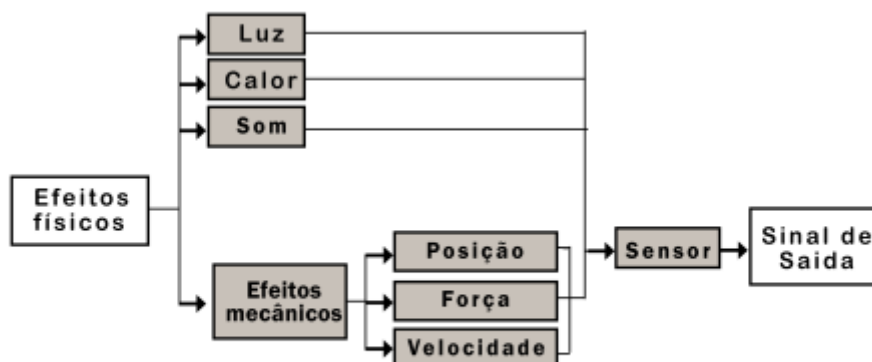
Fonte: Lopes et al (2018)

Já no modelo ativo os leitores não precisam fornecer tanta energia quando ocorre a comunicação e isso resulta em uma maior eficiência, tornando possível a comunicação para maiores distâncias. E como esses sistemas se aproveitam de funcionalidades mais complexas, faz-se necessário o uso de protocolos que necessitam de um maior poder de processamento (LOPES, PESTRE e TORRES , 2018).

## 2.4 Sensores

Sensores são dispositivos que respondem a um determinado estímulo ou sinal, esses sinais geralmente são uma forma de energia que pode ser luminosa, térmica, cinética, ou também relaciona a qualquer grandeza física que necessita ser medida como: temperatura, pressão, velocidade, corrente, etc (WENDLING, 2010). A Figura 10 apresenta as grandezas físicas que podem ser lidas por sensores:

Figura 10 - Formas de energia em um sensor



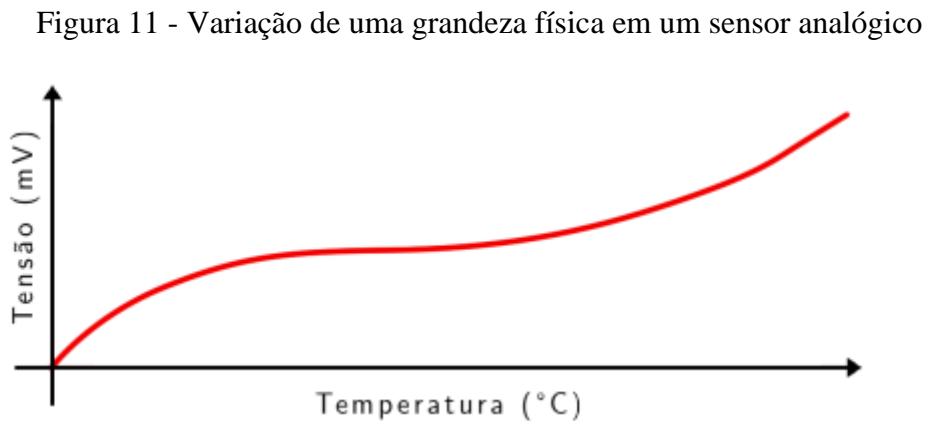
Fonte: Thomazini, Albuquerque (2020).

Nem todos os sensores possuem características elétricas que os possibilite serem utilizados diretamente no sistema de controle. Geralmente seu sinal de saída é manipulado por um circuito de interface para que possa ser lido pelo controlador e utilizado no sistema de controle (WENDLING, 2010).

Quando a saída do sensor é sensibilizada por alguma forma de energia, o seu sinal de saída pode ser um nível de tensão muito baixo e, nesses casos, é necessário a utilização de um amplificador capaz de elevar o seu nível de tensão para um valor adequado para a utilização (THOMAZINI e ALBUQUERQUE, 2020).

#### **2.4.1 Sensores analógicos**

São sensores que podem assumir quaisquer valores em seu sinal de saída ao longo do tempo, se trabalhando dentro de sua faixa de operação. Existem várias grandezas físicas que podem assumir qualquer valor ao longo do tempo (pressão, velocidade, temperatura, umidade, etc) (THOMAZINI e ALBUQUERQUE, 2020). A Figura 11 apresenta a variação da temperatura ( $^{\circ}\text{C}$ ) em um determinado sensor e sua tensão de saída em mV.



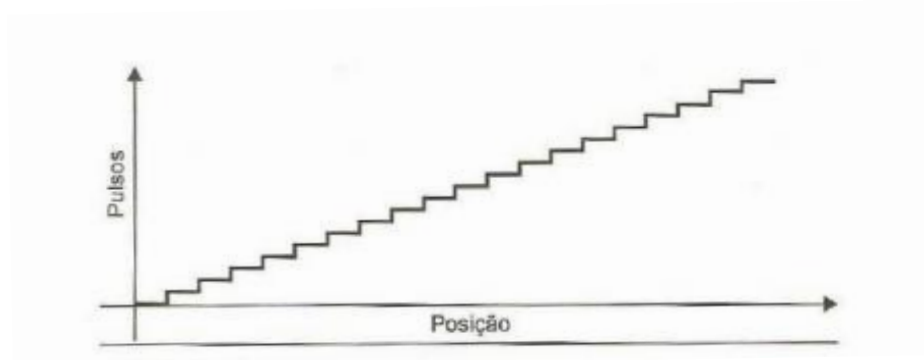
Fonte: Thomazini e Albuquerque (2020).

#### **2.4.2 Sensores digitais**

Os sensores digitais somente podem apresentar dois sinais de saída ao longo do tempo, sendo estes um ou zero (ligado/desligado). Não existem grandezas físicas que apresentam apenas esses dois valores, mas esses são vistos no sistema de controle após serem convertidos por algum circuito eletrônico (pressostato, termostato, e chave de nível), ou transdutores

(*encoders*) ou por códigos binários (BCD *Gray*, etc) (THOMAZINI e ALBUQUERQUE, 2020). A Figura 12 ilustra a variação da posição de um objeto por um *encoder* incremental.

Figura 12- Leitura da posição de um objeto por um *encoder* incremental



Fonte: Thomazini e Albuquerque (2020).

## 2.5 Linguagem *Python*

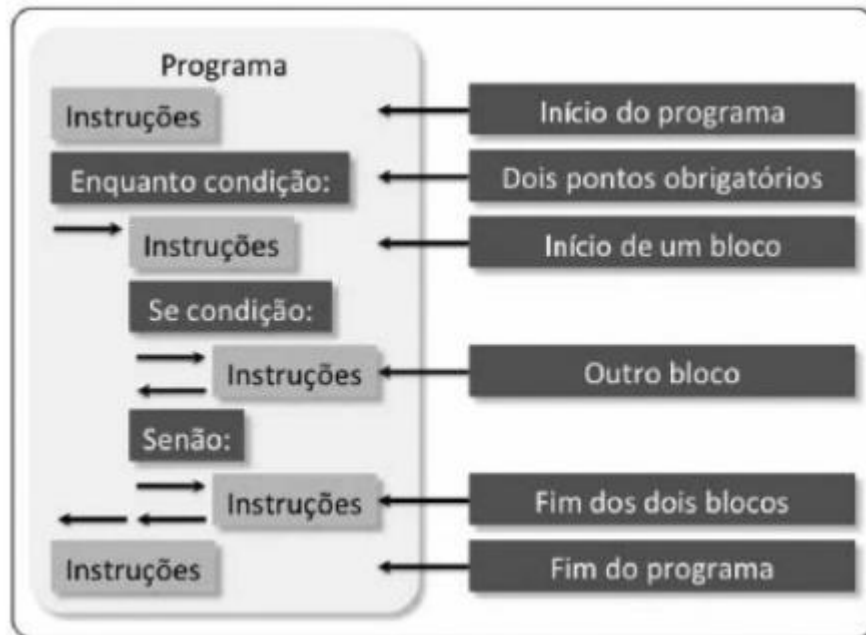
Neste trabalho, a linguagem de programação é utilizada para realizar a integração entre os dados dos sensores e a sua disposição em página *web*. Para tal, foi escolhida uma linguagem de programação que seja aceita pelo controlador utilizado e que também possua grande quantidade de referencial disponível.

A linguagem *Python* é uma linguagem livre e de alto nível orientada a objeto, de tipagem dinâmica e forte, ou seja, um dado não pode ser tratado como se fosse de outro tipo. Ela possui uma sintaxe clara e concisa que favorece sua legibilidade, sendo, por esse motivo, mais produtiva (BORGES, 2014).

Essa linguagem foi criada em 1990 por Guido van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI) e teve como foco inicial aplicações relacionadas à Física e à Engenharia. Ela foi desenvolvida a partir de uma linguagem já existente chamada ABC (BORGES, 2014).

Enquanto outras linguagens de programação utilizam marcações como ponto e vírgula (;), e ponto (.) para determinar o fim de cada linha de comando, além dos marcadores de blocos ({}), ou as palavras especiais (*begin/end*) (MENEZES, 2010). No *Python* cada linha representa uma nova linha de comando, mas também pode ser continuada em linhas seguintes utilizando a barra invertida (\). Os blocos são definidos de maneira mais simples por sua endentação, a Figura 13 apresenta de maneira clara a divisão de blocos.

Figura 13 - Estrutura de programa típico em Python



Fonte: BORGES (2014).

## 2.6 Sites em redes

Foi necessário o desenvolvimento de um *website* para disposição dos dados de maneira clara, e com a facilidade de acessar este por qualquer outro dispositivo na rede o site deve ser de fácil acesso.

Os serviços e facilidades de redes são requisitados por usuários através dos *softwares*, cada aplicação é basicamente um programa aplicativo em um computador se comunicando com outro programa aplicativo que roda em outro computador. Os serviços de rede abrangem um conjunto variado de atividades (COMER, 2016).

Nesse trabalho foi proposto a utilização do *Raspberry PI* como servidor para os dados do projeto, e também a utilização de seu chip *Wi-Fi* integrado para disposição desses dados em rede local, também será necessária a configuração e criação do banco de dados para armazenar e tratar os dados, bem como um site para melhor visualização dos dados. Nesse quesito, será desenvolvido um LAMP (*Linux, Apache, MySQL, PHP*), e para a criação do site serão utilizadas as linguagens PHP, HTML E CSS.

### 2.6.1 PHP

O PHP é uma das linguagens mais utilizadas na *web*. Existem milhões de *sites* no planeta que utilizam a linguagem PHP. Sua maior diferença quando comparado a outras linguagens é sua capacidade de interagir com o mundo *web*, deixando *websites* muito mais dinâmicos que permitem constantes mudanças no seu conteúdo sem a necessidade de alteração do código fonte por algum profissional da área (NIEDERAUER, 2017).

O PHP significa *Hypertext Preprocessor* e, originalmente, foi nomeado de *Personal Home Page Tools*. É uma linguagem de criação de *scripts* do lado do servidor, que pode ser incorporada ao HTML (CONVERSE e PARK, 2002)

Dentre as principais vantagens do PHP é possível citar o fato de essa ser uma linguagem gratuita, e que todo seu arquivo de instalação pode ser obtido no site oficial da linguagem. Além disso é uma linguagem com código fonte aberto e essas informações também podem ser encontradas no site oficial (NIEDERAUER, 2017).

### 2.6.2 HTML

O HTML é uma sigla para *HyperText Markup Language* ou em português Linguagem de marcação de hipertexto. Ela também é componente base da *web*, sendo responsável por construir *websites*, inserir novos conteúdos, como imagens e vídeos, por meio de hipertextos (TOTVS, 2023).

Os hipertextos são a união de diversos elementos que, quando interligados, estabelecem o site, permitindo comunicação, armazenamento e compartilhamento de informações. Nos sites existem diferentes formatações, parágrafos, *bullets* e fontes, e toda essa estruturação ocorre graças ao HTML (TOTVS, 2023).

Cada página dos sites é composta por *tags* de HTML que, em suma, são códigos que orientam a estrutura do documento, seu tamanho, fonte ou quebra de linhas. A estrutura e hierarquização das informações ocorre devido a essas marcações (TOTVS, 2023). Na lista abaixo serão descritos alguns exemplos de *tags* HTML:

- `<head>`: define o início do HTML;
- `<title>`: indica o título;
- `<body>`: é usado para demonstrar o início do conteúdo;
- `<h1>`, `<h2>` e `<h3>`: *tags* utilizadas para indicar títulos e subtítulos;
- `<strong>`: acrescenta o negrito ao texto;

- *<italic>*: sinaliza a formatação do conteúdo em itálico;
- *<p>*: insere o parágrafo,
- *<a href='LINK'>*: tag usada para inserir um link.

Desde a invenção da *World Wide Web* por Tim Berners-Lee, o HTML evoluiu por sete versões, que são (SILVA, 2014):

- HTML ;
- HTML +;
- HTML 2;
- HTML 3;
- HTML 3.2;
- HTML 4.0;
- HTML 4.01.

### 2.6.3 CSS

O CSS, sigla para *Cascading Style Sheets* ou em português, folha de estilo em cascata, é um recurso muito poderoso para formatação de páginas HTML. O HTML tem a função de inserir o conteúdo do site e o CSS atua como a “maquiagem” do *site*. Em suma, é capaz de realizar todos os processos referentes a formatação (MELO, 2015).

Este possui sua sintaxe dividida em três partes:

- Seletor – parte responsável por indicar qual *tag* HTML receberá o valor de formatação, também poderá ser selecionado um ID ou classe;
- Propriedade – é o atributo relacionado a *tag*, como exemplo podemos associar na *tag* (body), a propriedade *background*;
- Valor – é o valor atribuído à propriedade, tomando o exemplo acima poderíamos adicionar o valor verde para o *background* (MELO, 2015).

Os códigos CSS podem ser adicionados aos documentos de três formas, *in-line*, interno e externo (MELO, 2015).

## 2.7 Banco de dados

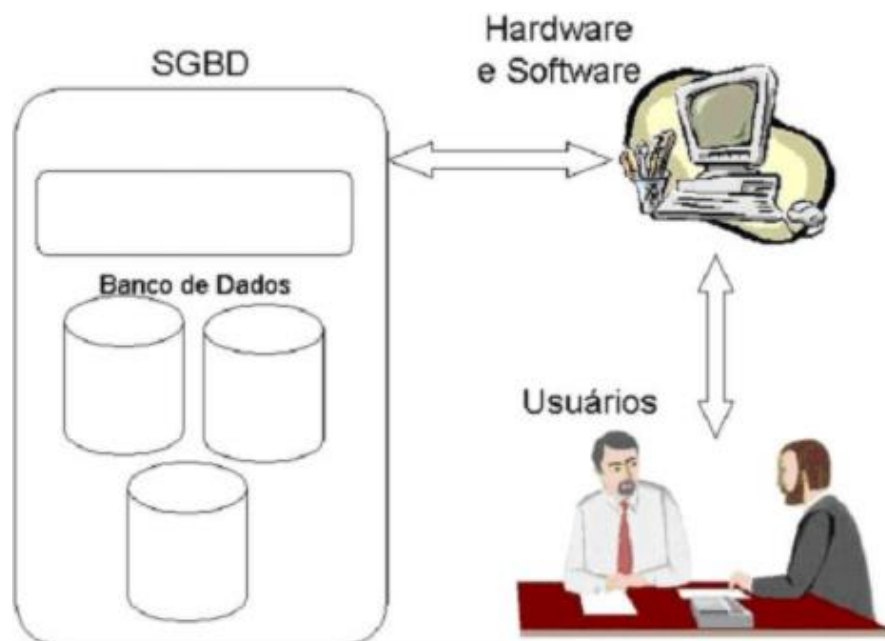
Os sistemas de banco de dados (SGBD) e os bancos de dados são ferramentas importantíssimas no dia a dia das pessoas. Todos os dias milhões de pessoas lidam com processos que dependem da interação com bancos de dados. Um exemplo é ir ao banco para retirar ou depositar dinheiro, acessar catálogos para diversos objetos, ou entrar em sites para aquisição de produtos (ELMASRI e NAVATHE, 2005).

Um banco de dados pode ser descrito genericamente como uma coleção de dados relacionados, que podem ser gravados e possuem significado implícito. Um banco de dados pode ser de qualquer tamanho e de complexidade variável (ELMASRI e NAVATHE, 2005).

O sistema gerenciador de banco de dados é a coleção de programas que permite aos usuários criar e manter bancos de dados além de proteger e garantir a manutenção dos bancos de dados por longos períodos (ELMASRI e NAVATHE, 2005). Alguns exemplos de SGBDs são: Oracle, SQL Server, MySQL, etc (REZENDE, 2023).

Os SGBDs então são *softwares* capazes de manipular as informações dos bancos de dados e também interagir com o usuário. Eles podem ser conceituados como o conjunto de quatro componentes básicos: dados, *hardware*, *software* e usuários (REZENDE, 2023). A Figura 14 ilustra os componentes dos bancos de dados.

Figura 14 - Componentes de um sistema de banco de dados



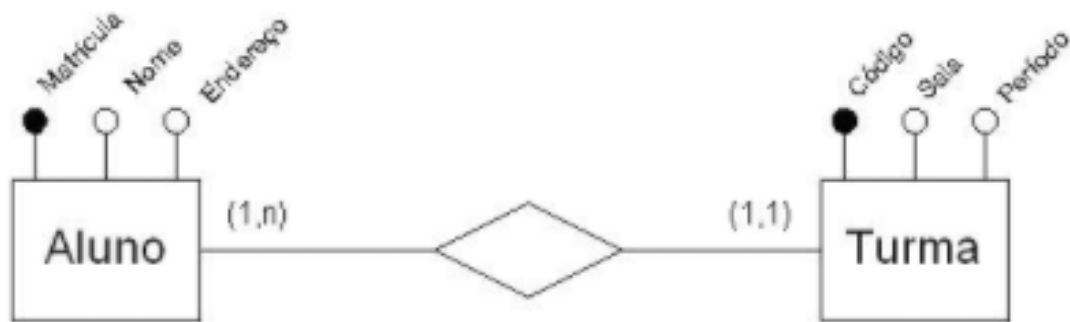
Fonte: REZENDE (2023).

Um sistema de banco de dados tem como objetivo promover a abstração de dados deixando o usuário isolado de detalhes internos, e torna-lo independente da aplicação, da estratégia de acesso e da forma de armazenamento (REZENDE, 2023).

Ao se projetar um banco de dados, deve-se atentar à organização das informações e ao desempenho, além de usar técnicas que facilitem a operação e manutenção. Para isso podemos realizar dois modelos: o modelo conceitual e o modelo lógico (REZENDE, 2023)

O modelo conceitual é uma maneira de apresentar o banco de dados sem se preocupar com o SGBD ou em como será feita sua implementação, para isso utiliza-se uma técnica que aborda a entidade-relacionamento (ER) (REZENDE, 2023). Na Figura 15 é possível observar o DER (diagrama entidade- relacionamento).

Figura 15 - DER (Diagrama Entidade- Relacionamento)



Fonte: REZENDE (2023).

A partir do modelo da Figura 15 conclui-se que, para cada aluno, será associado uma matrícula, um nome e um endereço e, para cada turma, serão armazenados código, sala e período.

No modelo lógico, o tipo de SGBD define como serão tratados os modelos lógicos, que podem ser relacionais, orientados a objetos, entre outros. Entretanto, o modelo mais difundido é o relacional, onde os dados são organizados em tabelas (REZENDE, 2023).

Figura 16 - Exemplos de tabelas em SGBD relacional

Aluno		
mat_aluno	nome	endereco
1	Cecília Ortiz Rezende	Rua dos Ipês, 37
2	Abílio José Dias	Avenida Presidente Jânio Quadros, 357
3	Renata Oliveira Franco	Rua Nove de Julho, 45

Turma		
cod_turma	sala	periodo
1	8	Manhãnte
2	5	Noite

Fonte: REZENDE (2023).

De acordo com as tabelas da Figura 16, é possível definir os nomes das colunas que compõe as tabelas e desenvolver o modelo lógico:

*Aluno (mat\_aluno, nome, endereço)*

*Turma (Cod\_turma, sala, período)*

Contudo, nesse modelo lógico, não são descritos os detalhes do armazenando, uma vez que esses dependem unicamente do *software* escolhido para implementar (REZENDE, 2023).

### 3 MATERIAIS E METÓDOS

Neste capítulo, são apresentados os equipamentos utilizados e as etapas principais de sua utilização, no protótipo do sistema de automação do controle leiteiro. O objetivo deste protótipo é armazenar dados, e apresentá-los de maneira adequada em uma rede local, criada pelo próprio *Raspberry Pi*.

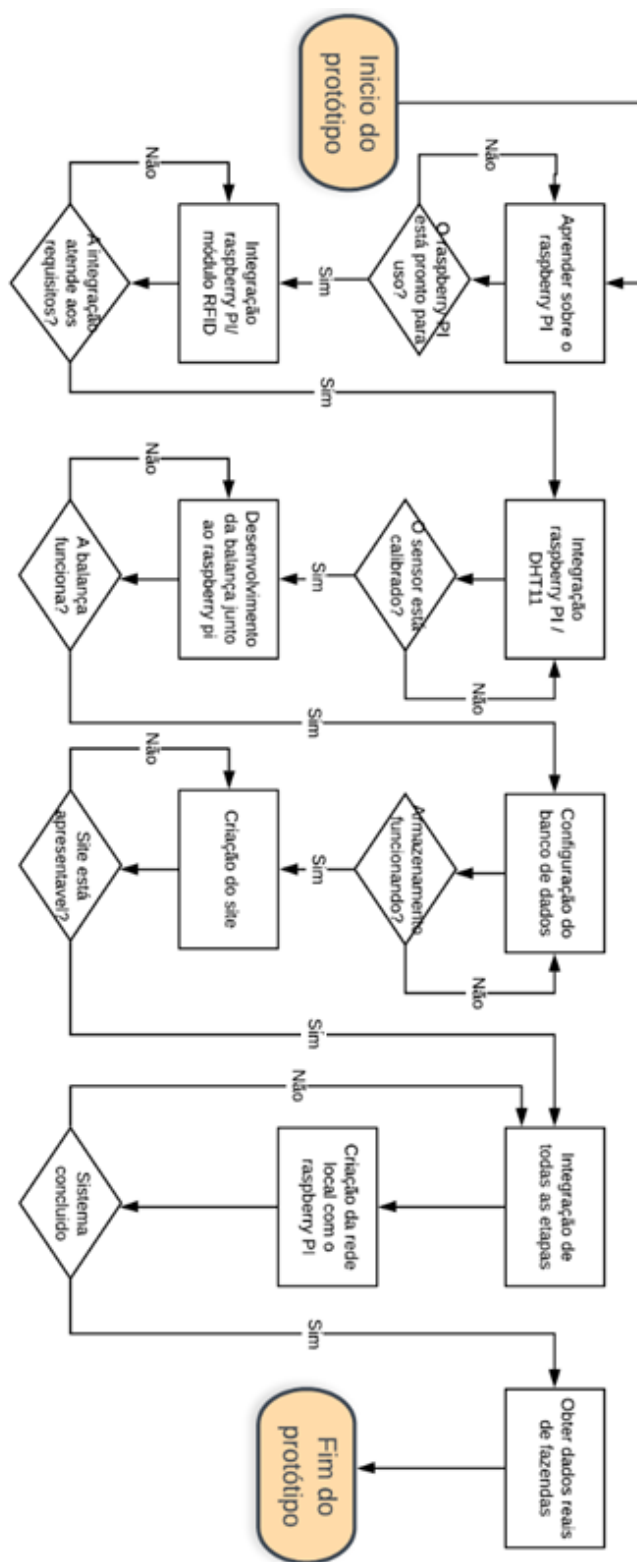
Esse protótipo foi montado com os seguintes equipamentos:

- Um *Raspberry PI* model B+;
- Um Módulo RFID MFRC522;
- 13 Chaveiros(*tag*) do tipo *Mifare*;
- Um Sensor de temperatura e humidade DHT11;
- Quatro células de carga de 20 kg;
- Um Módulo amplificador HX711;

Para apresentar os dados foi desenvolvido um *website*, utilizando das linguagens PHP, HTML E CSS.

Como o projeto no início era bastante complexo e envolvia conhecimento em diversos componentes, equipamentos e linguagens, optou-se por dividir o desenvolvimento do protótipo em partes, e ao fim uni-las e realizar os ajustes necessários. O fluxograma da Figura 17, ilustra melhor essa divisão:

Figura 17 - Fluxograma de desenvolvimento do protótipo



Fonte: Elaborado pelo autor (2021).

### 3.1 Raspberry Pi

O *Raspberry Pi* é um microcomputador que, apesar de possuir o tamanho de um cartão de crédito, também é capaz de executar os mais variados tipos de algoritmos e processamentos de dados. Nele é possível implementar *softwares* capazes de controlar seus pinos de entrada e saída.

Ele foi criado pela *Raspberry Pi Foundation* visando estimular o ensino da ciência da computação em escolas e universidades. Todo seu *hardware* é construído em uma placa de tamanho reduzido, mas, apesar do tamanho, possui um potencial de processamento significativo. O Arduino geralmente é comparado com o *Raspberry Pi*, mas este possui um sistema operacional rodando, apesar de não poder ser comparado com um computador tradicional pois possui diversos conectores acessíveis a criadores (CROTTI, SILVA, *et al.*, 2021).

Nesse projeto, a escolha do *Raspberry Pi* se deve a familiaridade com a linguagem de programação *Python* e também com a facilidade de acesso do *Raspberry Pi* a redes locais, afinal, diferente do Arduino, o *Raspberry* não necessita de nenhum *shield* para conectar ou criar sua própria rede local.

A *Raspberry Pi Foundation* lançou diversas versões do seu minicomputador, diferentes em relação a capacidade de processamento ou quantidade de pinos. Neste trabalho, optou-se por utilizar o *Raspberry Pi 3 model B+*, uma versão mais recente que apresenta grande poder de processamento, além de um adaptador *Wi-Fi*. A Tabela 3 apresenta as especificações técnicas desse modelo.

Tabela 3 - Especificações técnicas do *Raspberry Pi 3 B+*.

Modelo	<i>Raspberry Pi 3 Model B+</i>
Processador	<b>Broadcom BCM2837B0 64bits ARM Cortex-A53 Quad-Core</b>
<i>Clock</i>	1,4 GHz
Memória RAM	1 GB
Adaptador <i>WiFi</i>	802,11 b/g/n/AC 2.4 GHz e 5 GHz integrado
<i>Bluetooth</i>	4,2 BLE integrado
Conector de vídeo	HDMI

---

Conector <i>Ethernet</i>	Até 300 Mbps
Alimentação	Fonte DC 5V 3 A
Interface para câmera	(CSI)
–Interface para <i>display</i>	(DSI)
Slot para cartão microSD	-
– Conector de áudio e vídeo	-
GPIO	40 pinos
Dimensões:	85 x 56 x 17 mm

---

Fonte: *Raspberry Pi foundation*, 2023

A Figura 18 apresenta o *Raspberry Pi 3 Model B+*, que foi utilizado no sistema de automatização descrito nesse trabalho:

Figura 18 - *Raspberry Pi 3 Model B+*



Fonte: *Raspberry Pi foundation* (2023).

Como dito anteriormente, o *Raspberry Pi* é um computador e por isso necessita de um sistema operacional rodando para execução das tarefas, e para isso foi realizada a formatação do *Raspberry Pi*. A versão utilizada pode ser encontrada no site oficial da *Raspberry Pi Foundation*.

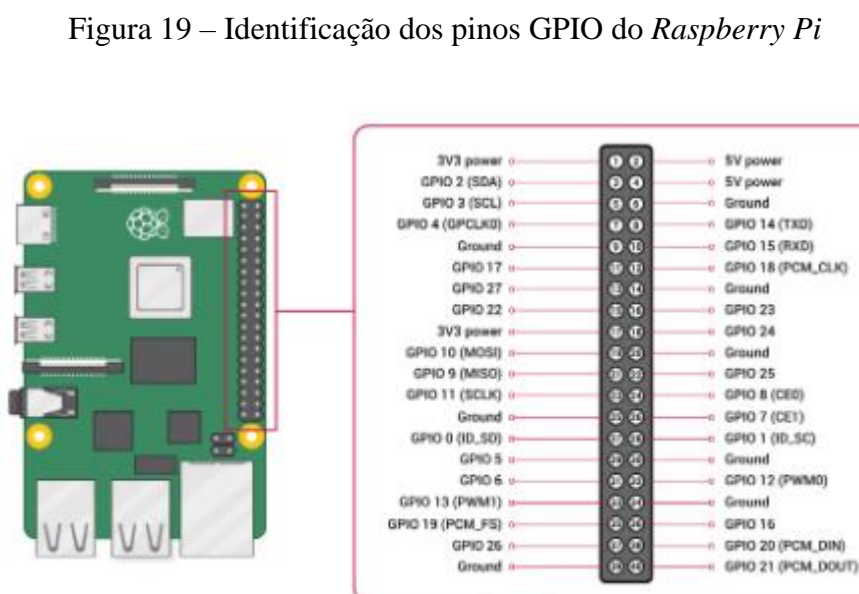
Existem duas versões disponíveis para *download*, a NOOBS (New Out Of the Box Software) e à NOOBS Lite. O NOOBS simplifica o processo, oferecendo uma interface amigável que permite aos usuários, escolher e instalar diferentes sistemas operacionais com facilidade. A versão NOOBS Lite, por sua vez, proporciona uma abordagem mais enxuta, exigindo uma conexão à internet para realizar o download dos sistemas operacionais selecionados. Em cada um desses *boots* de instalação livres existem diversos sistemas operacionais, contudo, para esse projeto, optou-se pelo *raspbian*.

Após o *download* e descompactação dos arquivos, estes foram salvos no cartão SD que será utilizado no *Raspberry Pi*. Na primeira instalação, o processo de instalação foi intuitivo e, após escolher o sistema a ser instalado e confirmar a seleção, esse processo foi concluído.

O sistema implementado conta com o compilador para Python já incorporado. Entretanto, os demais pacotes ou bibliotecas exigidos para o presente protótipo serão especificados conforme a necessidade. A atualização dessas bibliotecas será realizada de acordo com os requisitos dos equipamentos ou ferramentas exigidos.

Todo o projeto foi desenvolvido através do VNC *viewer*, um *software* livre que permite ao usuário utilizar o *Raspberry Pi* através de outro dispositivo.

A Figura 19 apresenta a identificação dos pinos General Purpose Input/Output (GPIO), estas são portas programáveis de entrada e saída de dados que são utilizadas para prover uma interface entre os periféricos e o *Raspberry Pi*.



Fonte: *Raspberry PI foundation*, 2023

### 3.2 Módulo MFRC522

No desenvolvimento do protótipo, foi utilizado um módulo baseado no chip MFRC522, da empresa NXP. Este chip possui baixo consumo e pequeno tamanho, além de permitir a leitura e escrita em cartões que possuem o padrão *Mifare*, que é um padrão que opera na frequência de 13,56 MHz, utilizado geralmente para distâncias entre 5-10 cm. A Figura 20 apresenta o módulo:

Figura 20 - Módulo leitor RFID MFRC522



Fonte: NXP, 2007.

Suas especificações técnicas podem ser vistas na Tabela 4:

Tabela 4 - Especificações técnicas módulo leitor *RFID MFRC522 Mifare*

Descrição	Valor
Corrente de trabalho	13-26 mA / DC 3.3 V
Corrente ociosa	10-13 mA / 3.3 V
Corrente <i>Slep</i>	<80 uA – Pico de corrente: <30 mA
Frequência de operação	13,56 MHz
Tipos de cartões suportados	<i>Mifare1 S50, S70 Mifare1, Mifare UltraLight, Mifare Pro, Mifare Desfire</i>
Temperatura de operação	-20 a 80 graus Celsius

---

Temperatura ambiente	-40 a 85 graus Celsius
Umidade relativa	5 % – 95 %
Parâmetro de Interface SPI	-
Taxa de transferência	10 Mbit/s
Dimensões	8,5 x 5,5 x 1,0 cm
Peso	21 g

---

Fonte: NXP, 2007.

As *tags* utilizadas foram do modelo *Mifare* tipo chaveiro e possuem a mesma frequência do módulo, distancia de leitura entre 3 e 5 cm e suas dimensões são 40,5 x 32 x 4,2 mm. A Figura 21 ilustra o modelo utilizado.

Figura 21 - *Tag* chaveiro RFID



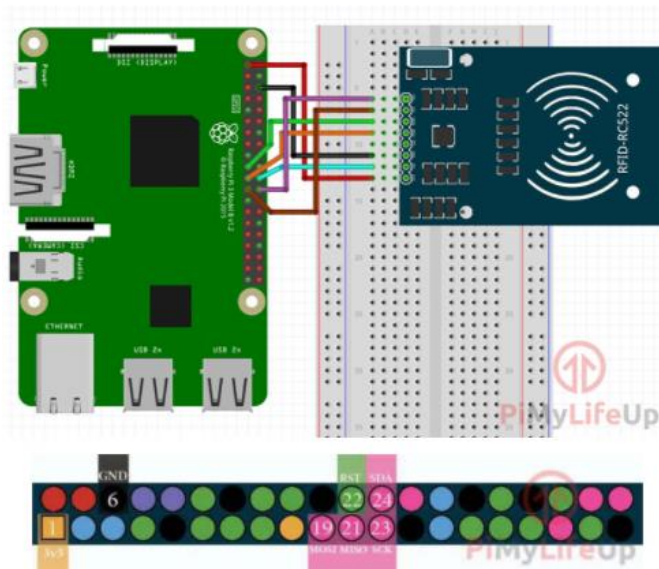
Fonte: NXP, 2007.

Para a conexão do *Raspberry* com o RFID MRFC522, foram utilizados 7 dos pinos GPIO do *Raspberry*, eles foram conectados diretamente ao módulo. A lista abaixo e a Figura 22 explicam a conexão entre os pinos do *Raspberry* e o módulo:

- SDA conectado ao pino 24;
- SCK conectado ao pino 23;
- MOSI conectado ao pino 19;

- MISO conectado ao pino 21;
- GND conectado ao pino 6
- RST conectado ao pino 22;
- 3,3 V conectado ao pino 1;

Figura 22 - Esquema de conexão para o módulo MRFC522



Fonte: Gus (2023).

Também foi necessário a instalação das ferramentas *python3-dev*, *python pip* e os pacotes *git*. O comando utilizado para essa instalação é apresentado na Figura 23:

Figura 23 - Comando para instalação das ferramentas necessárias

```
Terminal $
sudo apt-get install python3-dev python3-pip
```

Fonte: Gus, 2023.

Utilizando a ferramenta *pip* foi possível instalar a biblioteca *spidev*, e a biblioteca específica para o RFID RC522. Elas foram instaladas com os comandos descritos na Figura 24:

Figura 24 - Comando para instalar bibliotecas necessárias

```
Terminal $
sudo pip3 install spidev

Terminal $
sudo pip3 install mrfc522
```

Fonte: Gus, 2023.

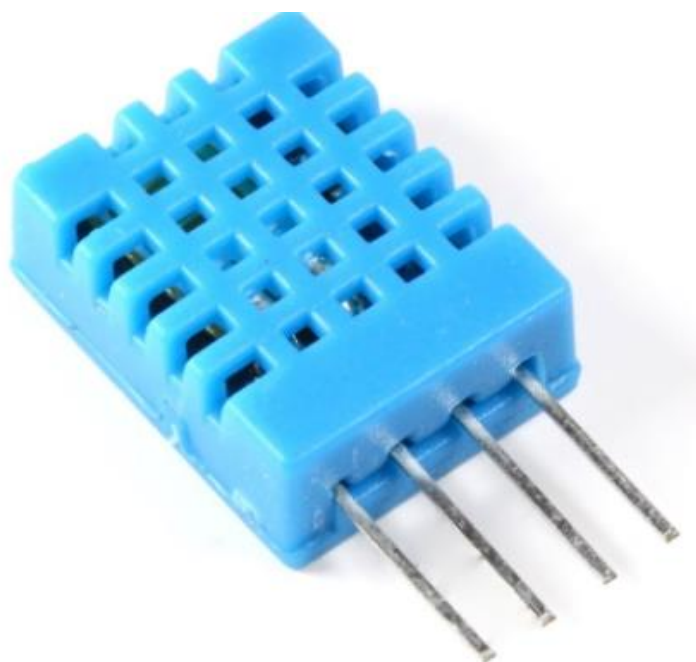
Foram desenvolvidos 2 programas, sendo um denominado ler.py e o outro escrever.py, disponíveis no Anexo A para visualização. O programa ler.py realiza a leitura do valor associado a cada *tag*, enquanto que o programa escrever associa algum valor para determinada *tag*.

### **3.2 DHT11**

O DHT11 é um sensor digital de humidade e temperatura, básico, e de baixo custo. Ele utiliza um sensor capacitivo de humidade e um termistor para medir a temperatura em seu entorno. Apresenta na saída um sinal digital no seu pino data, sem necessitar de um pino analógico ou de um conversor analógico/digital (SUNROM TECHNOLOGIES, 2021).

O sensor enfrenta uma limitação, visto que não é capaz de gerar dados em intervalos inferiores a 2 segundos (SUNROM TECHNOLOGIES, 2021). Tal limitação não compromete as análises pertinentes a este trabalho, uma vez que o período de aquisição é definido em relação ao tempo de ordenha, que é superior ao intervalo de 2 segundos. A Figura 25 ilustra o referido sensor.

Figura 25 - DHT11



Fonte: Sunrom Technologies (2021).

Suas especificações técnicas são vistas na Tabela 5:

Tabela 5 - Especificações técnicas DHT11

Descrição	Valor
Modelo	DHT11
Faixa de medição de umidade	20 a 90% UR
Faixa de medição de temperatura	0° a 50°C
Alimentação	3-5 VDC (5,5 VDC máximo)
Corrente	200 uA a 500 mA, em <i>stand by</i> de 100 uA a 150 uA
Precisão de umidade de medição	± 5,0 % UR
Precisão de medição de temperatura	2.0 °C
Tempo de resposta	2s
Dimensões	23 x 12 x 5mm (incluindo terminais)

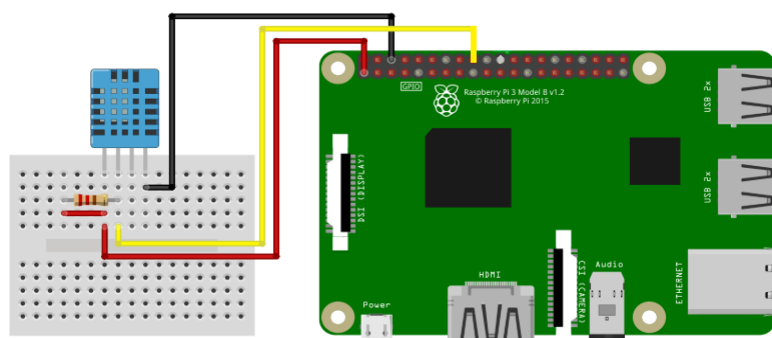
Fonte: Sunrom Technologies, 2021.

Foram utilizados 3 pinos do sensor DHT11, um para alimentação, um para aterramento e outro para recebimento de dados. Para segurança extra com o pino GPIO 24 foi utilizado um resistor *pull-up* de 4,7 k $\Omega$ . No experimento a seguir a conexão dos pinos utilizada é evidenciada na

Figura 26:

- VCC conectado ao pino 17;
- Ground conectado ao pino 14;
- Data conectado ao pino 18 (GPIO 24).

Figura 26 - Esquema de conexão para o DHT11



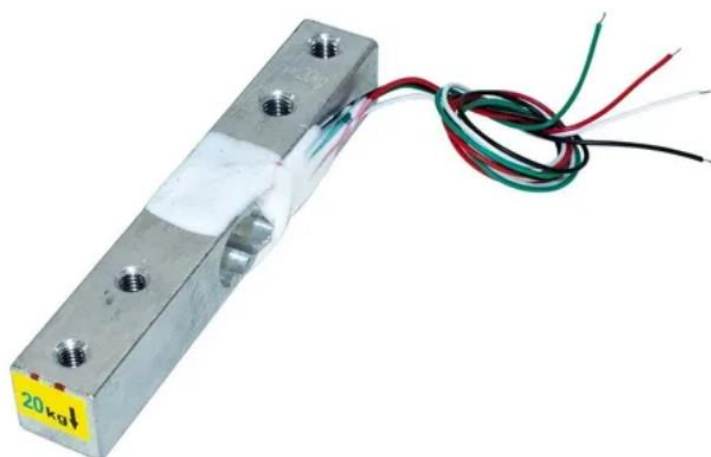
Fonte: Adilsom Thomsem (2015).

Como a ferramenta PI já foi instalada anteriormente, ela foi utilizada para instalar no *raspberry* a biblioteca *adafruit*, que atende o sensor DHT11 e também o DHT22. Também foi desenvolvido um programa capaz de coletar dados de temperatura e humidade ao ser executada. Seu código fonte pode ser encontrado no Anexo B e é nomeado DHT11.

### 3.3 Balança com células de carga e HX711

Neste trabalho foram utilizadas células de carga de 20 kg, do fabricante OEM, apresentada pela Figura 27:

Figura 27 - Célula de carga de 20 kg



Fonte: Voltriz (2023).

Na Tabela 6 são descritas as especificações técnicas para essa célula de carga.

Tabela 6 - Especificações técnicas célula de carga 20 kg.

Descrição	Valor
Modelo	Barra de alumínio
Carga nominal	0-20 kg
Potência nominal	1,0 ±0,1 mV
Zero saída	±0,1 mV
<i>Creep</i>	0,03 % f.s/30 min
Tensão de funcionamento	3-12 VDC
Tensão máxima de operação	15 VDC
Impedância de entrada	1066±10 % Ω
Impedância de saída	1000±10 % Ω
Temperatura de operação	-20-65 °C
Classe de proteção	IP65
Dimensões	11 mm (L)x 11 mm (A)x 80 mm (C)
Peso	30 g

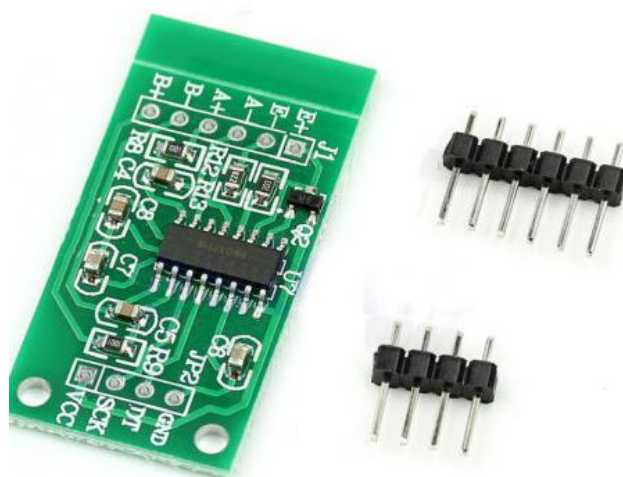
Fonte: Voltriz, 2023.

Na proposta deste trabalho, uma única célula de carga não suportaria o peso da lata de leite cheia, valor que chega a 50 kg, dependendo do volume da lata utilizada na fazenda. Para contornar o problema, foram utilizadas 4 células de 20 kg, afim de ter um peso máximo de 80 kg.

O HX711 é um conversor analógico digital de 24 bits e foi utilizado um módulo conversor para amplificar o sinal das células de carga e realizar a interligação entre estas e o microcontrolador. Ele é indicado para balanças, e também para controle de processos industriais (SEMICONDUCTOR, 2023).

Nesse trabalho foi utilizado o módulo já adaptado com o HX711, para trabalhar como conversor e amplificador. O módulo pode ser visualizado na Figura 28:

Figura 28 - HX711

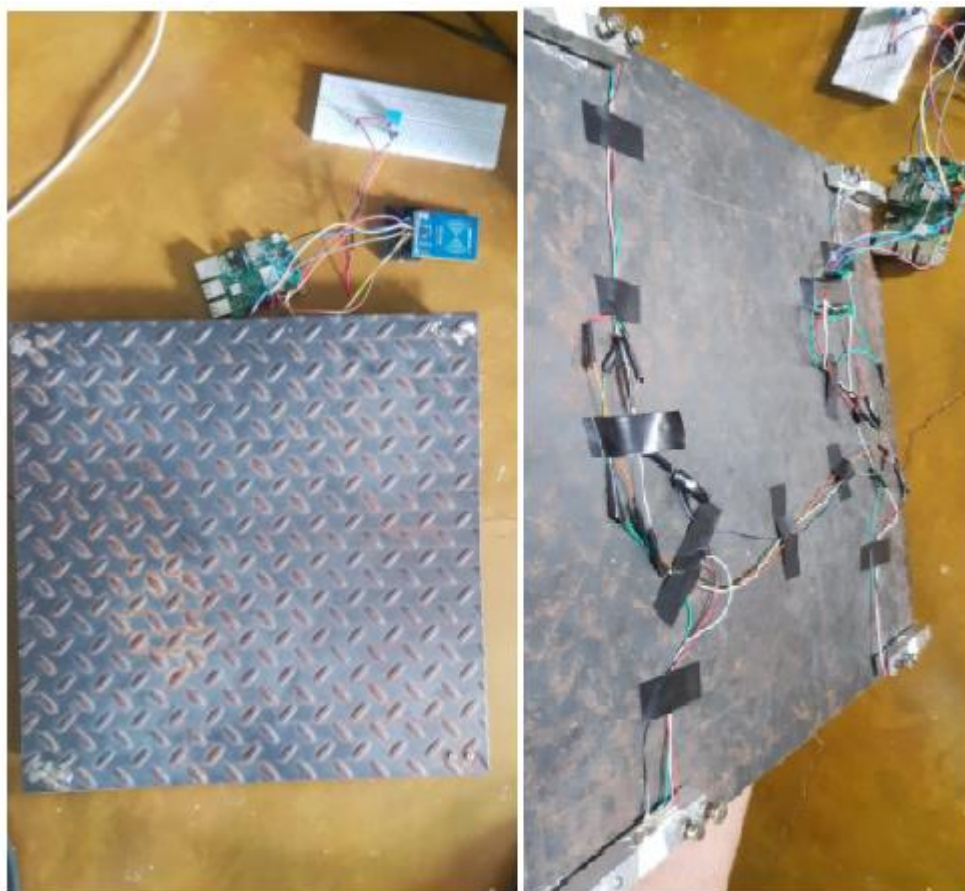


Fonte: Avia Semiconductor (2023)

O dispositivo possui uma tensão de operação de 4,8-5,5 VDC e uma corrente de operação de 1,6 mA, além de atuar em uma faixa de temperatura que vai de -20 °C até 85 °C e possuir interface SPI. Suas dimensões são de 29 x 17 x 4 mm sem os pinos (SEMICONDUCTOR, 2023).

Para desenvolver a balança, as células de carga de 20 kg foram dispostas na forma de um quadrado afim de que o peso aplicado fosse distribuído uniformemente. Elas foram fixadas em 2 placas de ferro com parafusos, e foram conectadas em paralelo. A Figura 29 apresenta a balança já montada:

Figura 29 - Balança desenvolvida



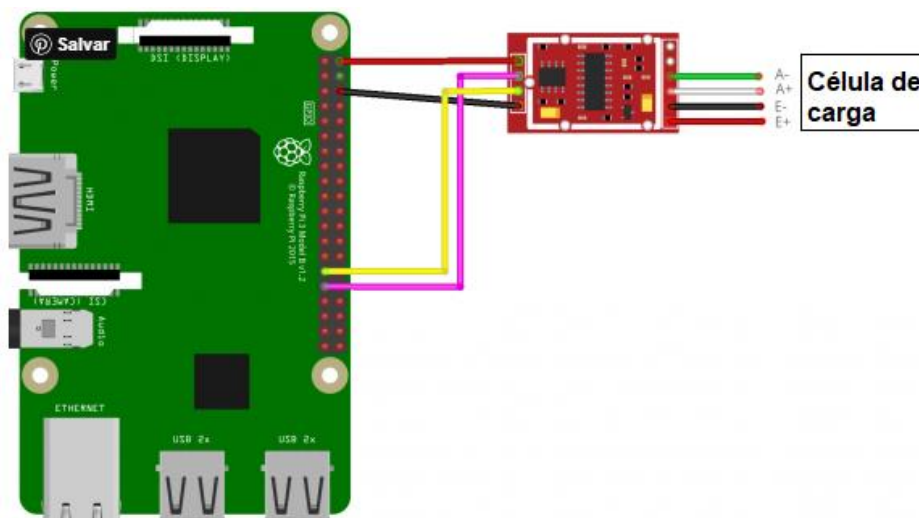
Fonte: Elaborado pelo autor (2022).

A conexão entre o HX711 e o *Raspberry Pi* foi estabelecida de acordo com a disposição esquemática apresentada na Figura 30, seguindo os seguintes detalhes:

- VCC do HX711 foi conectado ao pino 2 do *Raspberry Pi*;
- GND do HX711 foi conectado ao pino 6 do *Raspberry Pi*;
- DT do HX711 foi conectado ao pino 29 do *Raspberry Pi*;
- SCK do HX711 foi conectado ao pino 31 do *Raspberry Pi*.

Essa configuração específica garantiu uma conexão elétrica adequada entre os dispositivos, permitindo uma comunicação eficiente e confiável para a transmissão precisa de dados entre o sensor de peso HX711 e o *Raspberry Pi*.

Figura 30 - Ligação elétrica entre os objetos da balança

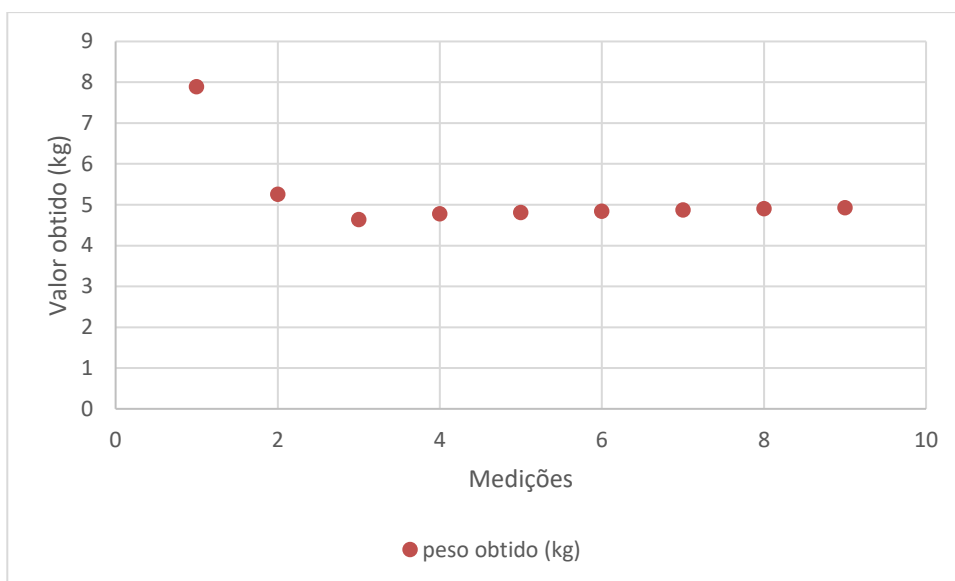


Fonte: Elaborado pelo autor (2022)

Por meio da ferramenta *Git*, adquiriu-se a biblioteca para o HX711, a qual inclui um exemplo denominado "example.py". Este código de exemplo foi utilizado para realizar testes após a conclusão da montagem da balança. O processo de calibração da balança iniciou-se com a determinação da tara, que representa o valor que estabelece a balança em 0, para isso o valor da constante “tara” foi determinado. Esse valor é impactado principalmente pela chapa de aço, que, mesmo sem carga adicional, exerce uma força sobre as células de carga. Posteriormente, foi essencial utilizar um objeto de peso conhecido para a calibração. A constante associada ao valor de calibração foi identificada como "referencia".

O valor da constante de “referencia” foi encontrado com o auxílio de uma balança de precisão. Durante esse procedimento, o valor de um peso padrão foi mensurado usando a balança de precisão. Através do método de comparação de resultados, a constante necessária para a calibração da balança foi determinada. Durante essa etapa, a resolução no programa foi configurada para dois dígitos decimais, visando identificar o valor mais preciso para a constante de calibração. A Figura 31 – Gráfico de calibração da balança Figura 31 ilustra a variação entre o peso obtido na balança desenvolvida (eixo y) e o número de cada medida (eixo x):

Figura 31 – Gráfico de calibração da balança



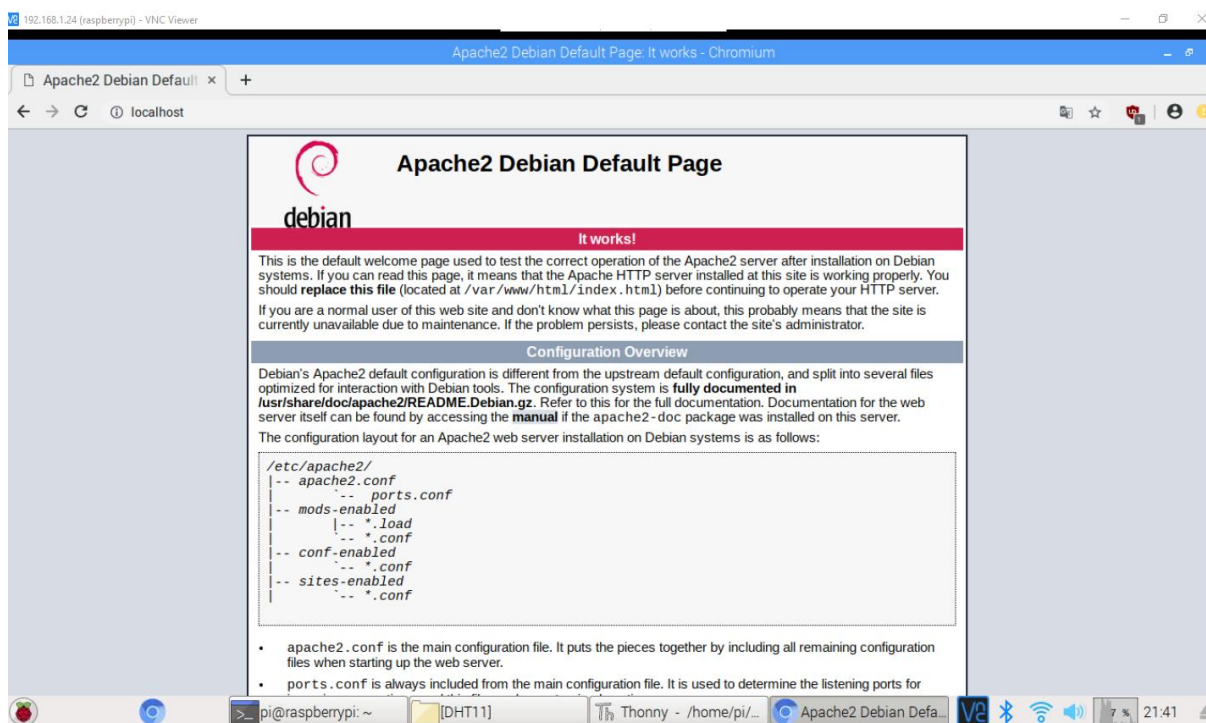
Fonte: Elaborado pelo autor (2023)

A análise do gráfico apresentado na Figura 31 revela que, mediante o ajuste da constante de calibração, os valores registrados pela balança convergem para uma maior proximidade com o valor de 4,933 kg obtido pela balança digital. Notavelmente, observa-se que o nono ponto de medição exibiu a maior proximidade em relação ao valor real. No entanto, resultados mais detalhados e uma avaliação mais aprofundada da confiabilidade da balança serão apresentados em uma seção subsequente do trabalho, proporcionando uma visão abrangente da precisão e consistência do instrumento de medição.

### 3.4 Criação do LAMP

A implementação do banco de dados requer a utilização de alguns *softwares* adicionais que incorporam linguagens essenciais para sua criação e manutenção. O servidor *web* utilizado foi o *Apache*, e pode ser adquirido no terminal do *Raspberry Pi* por meio do comando "`sudo apt-get install apache2 -y`". Para verificar sua instalação, basta digitar "`http://localhost`", e a tela inicial do *Apache* será exibida. A mudança para a tela inicial do *Apache*, apresentada pelo *VNC-viewer* após a instalação, é ilustrada na Figura 32.

Figura 32 - Tela inicial do apache apresentada pelo VNC-viewer após instalação

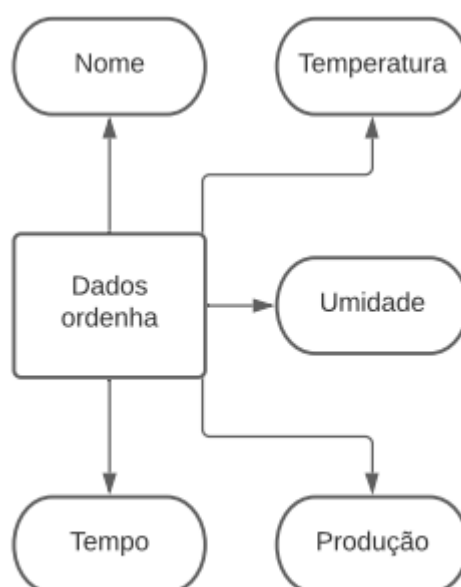


Fonte: Elaborado pelo autor (2022).

É relevante destacar que, ao inserir o endereço de IP (Protocolo de *Internet*) do *Raspberry* em qualquer navegador de um dispositivo conectado à rede, a página visualizada corresponderá à mencionada. Para a concepção de um *site* capaz de atualizar seus valores sempre que novos dados forem recebidos, torna-se indispensável a adoção do PHP, proporcionando dinamismo ao *site*. A instalação do PHP foi realizada por meio do comando "`sudo apt-get install php -y`", e a confirmação da instalação adequada foi verificada substituindo o arquivo `index.html`, que compreende a página inicial do servidor *Apache*, por um *script* PHP exibindo a data atual. O reinício do *Apache* foi necessário para o funcionamento do *script*.

A ferramenta empregada na criação do banco de dados foi o MariaDB, e todos os pacotes associados ao PHP e MySQL foram igualmente instalados através do comando "`sudo apt-get install mariadb-server php-mysql -y`". Para a elaboração do banco de dados, utilizou-se o *PhpMyAdmin*, uma ferramenta simplificada que facilita a criação de tabelas. Foi criada uma tabela única destinada ao armazenamento dos dados provenientes dos sensores e da balança. A estrutura da tabela é apresentada na Figura 33:

Figura 33- Estrutura da tabela criada



Fonte: Elaborado pelo autor (2022).

O banco de dados (BD) relacionado à ordenha compreende atributos como nome, temperatura, umidade, produção e tempo durante o processo de ordenha. Adicionalmente, incorpora um identificador (ID) com incremento automático para otimizar a organização dos dados. A Figura 34 ilustra a tabela gerada no *PhpMyAdmin*. Este banco de dados destina-se a armazenar informações cruciais para o monitoramento do processo de ordenha, fornecendo uma estrutura organizada para consulta e análise posterior.

Figura 34 - Tabela Criada no *Phpmyadmin*

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1	<b>Nome</b>	varchar(120) utf8mb4_general_ci		Não	Nenhum			Muda  Elimina  Mais
<input type="checkbox"/>	2	<b>Tempo</b>	int(48)		Não	Nenhum			Muda  Elimina  Mais
<input type="checkbox"/>	3	<b>Temperatura</b>	int(48)		Não	Nenhum			Muda  Elimina  Mais
<input type="checkbox"/>	4	<b>Umidade</b>	int(48)		Não	Nenhum			Muda  Elimina  Mais
<input type="checkbox"/>	5	<b>Producao</b>	int(48)		Não	Nenhum			Muda  Elimina  Mais
<input type="checkbox"/>	6	<b>ID</b>	int(48)		Não	Nenhum		AUTO_INCREMENT	Muda  Elimina  Mais

Fonte: Elaborado pelo autor (2022).

### 3.5 Raspberry PI como ponto de acesso

Para configurar o *Raspberry Pi* como um ponto de acesso, é essencial instalar dois programas específicos: *hostapd* e *dnsmasq*. No entanto, é necessário editar determinados arquivos de configuração desses programas, o que demanda a desativação prévia dos mesmos. Os comandos descritos na Figura 35 executam esse procedimento, permitindo a intervenção necessária nos arquivos de configuração para efetuar as alterações adequadas. Esse processo é vital para assegurar a correta configuração do *Raspberry Pi* como ponto de acesso, garantindo a funcionalidade desejada no contexto do projeto de automação leiteira.

Figura 35 - Comandos para obter/ parar os programas hostapd e dnsmasq.

```
sudo apt-get install hostapd
sudo apt-get install dnsmasq

sudo systemctl stop hostapd
sudo systemctl stop dnsmasq
```

Fonte: Elaborado pelo autor (2022).

Para estabelecer um endereço IP estático para o wlan0, é imprescindível realizar a edição do arquivo de configurações *dhcpcd*. Esse processo envolve a inclusão de comandos específicos no final do arquivo, a fim de garantir a configuração estática desejada. A necessidade de um IP estático para a interface wlan0 reside na estabilidade e previsibilidade do endereço IP, crucial para o correto funcionamento do projeto de automação leiteira. A inserção dos comandos pertinentes no arquivo *dhcpcd* permite personalizar as configurações de rede do *Raspberry Pi*, garantindo a funcionalidade do *Rasperry* como ponto de acesso. Os comandos descritos são observados na

Figura 36:

Figura 36 - Comandos para configurar um IP estático para a interface wlan0

```
interface wlan0
static ip_address=192.168.0.10/24
```

Fonte: Elaborado pelo autor (2022).

Para configurar o servidor DHCP no *dnsmasq*, o arquivo de configuração básico é excluído e um novo arquivo é criado com a seguinte configuração:

Figura 37 - Configuração do servidor DHCP

```
interface=wlan0
dhcp-range=192.168.0.11,192.168.0.30,255.255.255.0,24h
```

Fonte: Elaborado pelo autor (2022).

Neste estágio, é crucial configurar o *software* encarregado da criação do ponto de acesso, denominado *hostapd*. O arquivo de configuração deve refletir os ajustes indicados na Figura 38, onde os dados da rede são personalizados por meio dos parâmetros *ssid* e *wpa-passphrase*. Essas configurações desempenham um papel vital na definição das características específicas da rede *Wi-Fi*, garantindo uma integração adequada ao contexto do projeto de automação leiteira em questão.

Figura 38 - Configurações do ponto de acesso

```
interface=wlan0
bridge=br0
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
ssid=NETWORK
wpa_passphrase=PASSWORD
```

Fonte: Elaborado pelo autor (2022).

Outro arquivo que necessita de edição é o `sysctl.conf`. Neste arquivo, remova o caractere `"#"` da linha `"net.ipv4.ip_forward=1"`. Por fim, as regras da tabela de IP devem ser ajustadas conforme os comandos apresentados na Figura 39. Essas modificações são essenciais para garantir a configuração adequada do sistema, proporcionando uma integração eficaz dos elementos envolvidos no projeto em questão.

Figura 39 – Comandos para organizar a tabela de IP's

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"

iptables-restore < /etc/iptables.ipv4.nat
```

Fonte: Elaborado pelo autor (2022).

### 3.6 Desenvolvimento do *website*

Com o objetivo de aprimorar a acessibilidade aos dados e fornecer informações essenciais sobre o projeto aos usuários, criou-se um site composto por quatro páginas distintas. A página inicial, é dedicada à apresentação de informações relevantes sobre os equipamentos empregados no projeto. Outra página é reservada para compartilhar detalhes sobre o desenvolvedor do projeto, proporcionando uma visão mais pessoal e contextual. A quarta página, crucial para a compreensão do projeto, importa todos os dados coletados e os apresenta em formato tabular. Por último, destaca-se a página de gráficos, que permite aos usuários visualizarem de maneira efetiva como está se desenvolvendo a produção. As Figuras 38 a 41 oferecem uma ilustração visual de cada uma das páginas descritas.

Figura 40 - Página de detalhes do projeto



Fonte: Elaborado pelo autor (2022).

A página inicial do site destaca os principais componentes do projeto de automação: *Raspberry Pi*, sensor de temperatura e umidade (DHT11) e a balança eletrônica. Esses elementos são cruciais para o controle eficiente e a coleta de dados precisos sobre a produção de leite. O *Raspberry Pi* atua como o cérebro do sistema, enquanto o DHT11 fornece informações ambientais importantes. A balança, por sua vez, é essencial para monitorar e registrar a quantidade exata de leite produzido pelos animais. Essa disposição dos componentes pode ser visualizada na Figura 40.

Figura 41 - Página de dados do desenvolvedor



Fonte: Elaborado pelo autor (2022).

Como pode ser observado na Figura 41 a segunda página do site apresenta os mesmos menus simplificados apresentados anteriormente, proporcionando uma navegação fácil e acessível para conhecer mais sobre o projeto, explorar informações detalhadas sobre os

desenvolvedores, visualizar a disposição em gráficos e tabelas. Além disso, a página apresenta uma breve descrição dos desenvolvedores. Essa abordagem visa fornecer uma visão abrangente do projeto e garantir que os usuários encontrem facilmente as informações que desejam.

Figura 42 - Página de dados do animal Esmeralda disposto em tabela



Tabela de informações da Esmeralda

Dia	Tempo (s)	Temperatura (°C)	Umidade (%)	Produção (Kg)
1	210	25.1	37	10.7
2	205	25.5	40	10.9
3	190	24	40	11
4	198	26	40	9
5	195	27.1	41	9.8

Projeto desenvolvido no IFMG -campus formiga.

Fonte: Elaborado pelo autor (2022).

A terceira página apresentada na Figura 42 ilustra a mesma simplicidade dos menus, proporcionando aos usuários a possibilidade de alterar entre os menus. Um botão permite que os visitantes escolham o parâmetro de visualização das tabelas, oferecendo uma análise mais aprofundada da produção leiteira. Essa funcionalidade acrescenta uma camada de personalização, permitindo que os usuários explorem informações específicas de interesse com facilidade. Essa abordagem visa proporcionar uma experiência interativa e informativa aos usuários, atendendo às suas necessidades particulares de análise de dados.

A quarta e última página, conforme ilustrada na Figura 43, preserva a simplicidade dos menus anteriores, possibilitando a navegação entre as opções disponíveis. Um botão dedicado permite aos visitantes selecionar os parâmetros desejados para a visualização em gráficos, oferecendo uma análise mais detalhada e visualmente informativa da produção leiteira. Essa funcionalidade permite que os usuários explorem dados específicos de seu interesse com facilidade. Essa abordagem visa proporcionar uma experiência interativa e esclarecedora, atendendo às necessidades particulares dos usuários em relação à análise gráfica, que oferece uma compreensão visual mais intuitiva e aprofundada dos dados coletados.

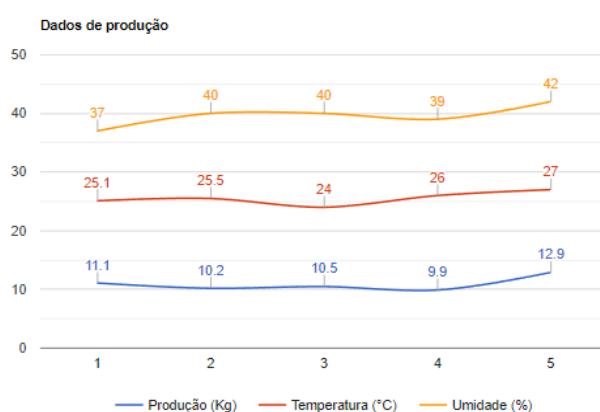
Figura 43 – Página dedicada a apresentação das informações em forma de gráficos



Selecione o animal para gerar o gráfico:

Selecione ▾

Enviar



Projeto desenvolvido no IFMG -campus formiga.

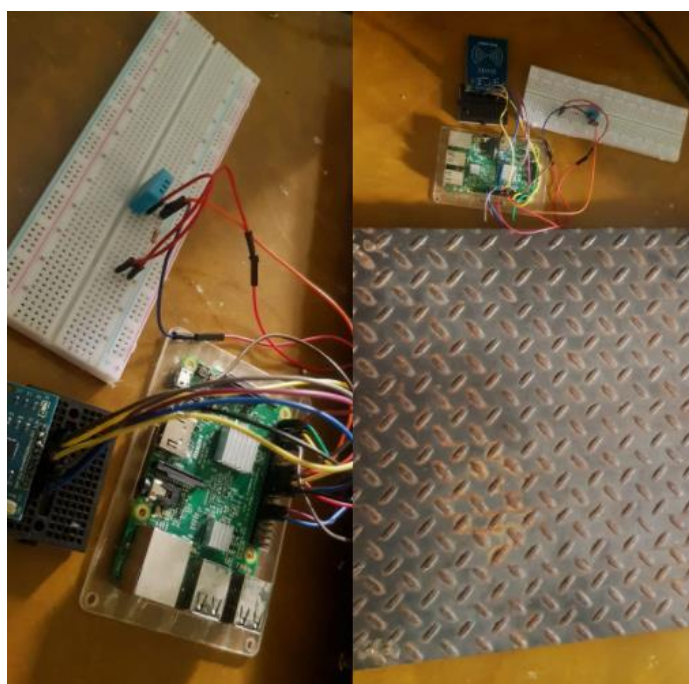
Fonte: Elaborado pelo autor (2022).



- *Ground* do DHT11 conectado ao pino 14;
- *Data* do DHT11 conectado ao pino 18 (GPIO 24).
- SDA do MFRC522 conectado ao pino 24;
- SCK do MFRC522 conectado ao pino 23;
- MOSI do MFRC522 conectado ao pino 19;
- MISO do MFRC522 conectado ao pino 21;
- GND do MFRC522 conectado ao pino 9
- RST do MFRC522 conectado ao pino 22;
- 3,3 V do MFRC522 conectado ao pino 1;

A Figura 45 destaca a interconexão entre a balança, o sensor DHT11 e o MFRC522 com o *Raspberry Pi*, evidenciando a integração eficaz desses componentes no sistema. A robustez da balança é notável, pois é projetada para suportar pesos de até 80 kg e para isso utiliza como material na superfície uma chapa de aço, proporcionando uma capacidade substancial de medição. Sua estrutura resistente e materiais de construção foram cuidadosamente selecionados para garantir durabilidade e precisão, atendendo assim aos requisitos específicos do projeto. Essa capacidade de carga significativa torna a balança adequada para aplicações na automação leiteira, onde a precisão e a resistência são essenciais para a coleta de dados confiável ao longo do tempo.

Figura 45 - Protótipo construído



Fonte: Elaborado pelo autor (2022).

O código fonte do programa responsável pela coleta de dados e seu armazenamento pode ser visualizado nos anexos, sob o nome de “coleta.py”.

### 3.8 Características da balança

Como visto na seção 2.2 deste trabalho, existem diversas terminologias associadas às medidas e, para descrever as capacidades e limites da balança, foi necessário realizar testes para determinar valores para essas medidas.

A resolução da balança foi determinada de acordo com a necessidade do projeto, e para fins de controle leiteiro, diferenças inferiores a 100 g não são relevantes. Essa resolução foi determinada diretamente na linguagem de código onde o valor final é apresentado com apenas uma casa decimal.

Para determinar a exatidão da balança foram realizadas 10 medidas em 3 pesos de prova massas 1,539 kg, 4,933 kg e 6,640 kg respectivamente. Estes valores foram comparados com uma balança de precisão que possui resolução de 1 g. Essa balança é mais do que suficiente para efeitos de comparação, pois para o propósito do projeto uma resolução de 100 g já é suficiente, afinal para ajustes de manejo no rebanho valores inferiores a 100 g são irrelevantes. Os valores das medidas podem ser observados na Tabela 7 :

Tabela 7 - Medições dos 3 pesos de prova na balança desenvolvida

Peso de prova 1 (kg)	Peso de prova 2 (kg)	Peso de prova 3 (kg)
1,5	4,9	6,6
1,5	4,9	6,6
1,5	4,9	6,6
1,5	4,9	6,6
1,6	4,9	6,6
1,6	5,0	6,6
1,6	5,0	6,6
1,6	5,0	6,6
1,6	5,0	6,6
1,6	5,0	6,6

Fonte: Elaborado pelo autor (2022).

A Tabela 8 – Resumo dos valores para determinar a exatidão Tabela 8 apresenta uma análise detalhada das medições realizadas pela balança em comparação com os valores obtidos pela balança de precisão. Cada linha destaca uma medida específica, indicando a média correspondente, o valor real registrado pela balança de precisão e a diferença entre esses dois, que representa a exatidão de cada medida. As diferenças individuais são apresentadas como 0,021 kg, 0,017 kg e 0,04 kg para as três medições, respectivamente. A média dessas diferenças, resulta em uma exatidão geral da balança de 0,026 kg. Esses dados fornecem uma compreensão abrangente da precisão do instrumento em relação às medidas realizadas.

Tabela 8 – Resumo dos valores para determinar a exatidão

<b>Identificação do peso de prova</b>	<b>Média de 10 medições (kg)</b>	<b>Valor real da balança de precisão (kg)</b>	<b>Exatidão (kg)</b>
Peso de prova 1	1,56	1,539	0,021
Peso de prova 2	4,95	4,933	0,017
Peso de prova 3	6,6	6,64	0,04

Fonte: Elaborado pelo autor (2023).

Utilizando-se dos valores obtidos também foi possível realizar o desvio padrão de cada uma das séries de medidas e determinar a precisão da balança. O desvio padrão pela equação 4:

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N}} \quad (4)$$

- $x_1, x_2, x_3$  são os dados envolvidos (quilograma -kg);
- $\mu$  é a média aritmética dos dados (quilograma -kg);
- $N$  é a quantidade de medidas.

O desvio padrão para cada série de medidas foi 0,052 kg para o peso de prova 1, 0,053 kg para o peso de prova 2 e 0 para o peso de prova 3. A média desses valores representa a precisão da balança que foi de 0,052 kg.

A sensibilidade da balança é definida como o menor incremento de medida que pode ser detectado pelo sistema. Neste contexto específico, a resolução da balança, representada pelo menor valor que ela pode mensurar, é de 100 gramas. Portanto, a sensibilidade dessa balança é equivalente à sua resolução, sendo também de 100 gramas. Isso implica que a balança é capaz de distinguir variações de peso no intervalo de 100 gramas, sendo esse o menor incremento que pode ser identificado em suas leituras.

### 3.9 Coleta de dados

A aquisição de dados foi conduzida em uma fazenda, onde foram monitorados 13 animais ao longo de um período de cinco dias, com ênfase na produção de ordenha matutina. Ressalta-se que as *tags* RFID utilizadas não contavam com proteção adicional contra intempéries. Cada *tag* foi numerada de 1 a 13, estabelecendo uma associação única com um animal específico. A Tabela 9 detalha a correlação entre as *tags* e os animais na mencionada fazenda.

Tabela 9- Relação *tag* e nome dos animais para a Fazenda

Número da <i>tag</i>	Nome do animal
1	Laranjinha
2	Paloma
3	Jurema
4	Pretinha
5	Nambuca
6	Pintada
7	Rainha
8	Esmeralda
9	Espadilha
10	Sete copas
11	Viola
12	Moirinha

Fonte: Elaborado pelo autor (2022).

Na

Figura 46, que valida o funcionamento do protótipo, é possível visualizar o sistema em pleno funcionamento, com destaque para a balança eletrônica integrada ao *Raspberry Pi*, o sensor de temperatura e umidade (DHT11) e o leitor RFID (MRFC522). A precisão e estabilidade da balança são evidenciadas pela representação gráfica dos dados, que indicam com precisão a quantidade de leite produzido.

A robustez do protótipo é comprovada pela capacidade de suportar o peso da lata de leite sem comprometer a precisão das medições. A imagem mostra claramente o equilíbrio da balança, refletindo a eficácia do sistema na coleta de dados de produção leiteira. Essa representação visual reforça a confiabilidade do protótipo em ambientes agrícolas, onde a precisão e a robustez são elementos cruciais para o monitoramento eficiente da produção. Vale ressaltar a importância de um *nobreak*, considerando a ausência de uma fonte de energia de fácil acesso no local.

Figura 46 - Protótipo em funcionamento no momento da ordenha



Fonte: Elaborado pelo autor (2022).

## 4 RESULTADOS E DISCUSSÕES

A primeira metade do presente projeto envolveu a abordagem individual dos equipamentos, visando compreender sua operação e a subsequente interconexão com o *Raspberry Pi*. Na segunda metade, ocorreu a integração de todas as partes do projeto, culminando na sua implementação em campo.

Durante a fase inicial da montagem e colocação em funcionamento do protótipo, não aconteceram dificuldades técnicas que acarretassem em atraso para o cumprimento destas atividades. Os equipamentos apresentaram desempenho conforme estavam preconizados em seus respectivos manuais. Um ponto crucial foi a necessidade de aprofundar o entendimento sobre o modelo de operação do *Raspberry Pi*.

Destaca-se que o *Raspberry Pi* é um computador portátil que opera em um sistema baseado em *Linux* (*Raspbian*), diferenciando-se de muitos computadores convencionais. O domínio dos sistemas *Linux*, menos intuitivos em comparação aos sistemas *Microsoft*, é essencial para trabalhar eficientemente com o *Raspberry Pi*.

No entanto, a *Raspberry Pi Foundation* oferece recursos valiosos, como artigos, que facilitam a iniciação com o *Raspberry Pi*, tornando possível a realização de diversos projetos mesmo para aqueles com conhecimento limitado em sistemas baseados em *Linux*.

A escolha da linguagem de programação *Python* mostrou-se bem-sucedida, permitindo o desenvolvimento eficaz de códigos de teste para os equipamentos. A alta legibilidade e a intuição da linguagem, aliadas a fóruns *online* explicativos, contribuíram para essa conquista.

A configuração do *Raspberry Pi* como servidor *web* envolveu a instalação sem problemas dos *softwares Apache*, *PHP*, *MySQL* (*MariaDB*) e *PhpMyAdmin*, conforme detalhado na seção 4.4. Não foram registrados problemas ou falhas durante esse processo.

A etapa mais desafiadora envolveu a modificação do adaptador Wi-Fi do *Raspberry Pi* para atuar como ponto de acesso. Nesse contexto, a alteração de parâmetros nos arquivos de sistema exigiu cuidado, com eventuais erros na sintaxe levando à necessidade de formatação do sistema. A criação do *site*, utilizando as linguagens mencionadas no capítulo 2, demandou tempo, requerendo um claro entendimento da sintaxe e lógica dessas linguagens.

A avaliação da balança desenvolvida revelou uma precisão de 0,026 kg, indicando que os valores medidos se desviam, em média, por esse montante em relação ao valor real. A precisão encontrada foi de 0,052 kg, representando o desvio padrão. Ambos os resultados se

situaram abaixo da resolução proposta para a balança e foram compatíveis com os fins dessa atividade.

Com os resultados iniciais positivos, a segunda etapa do projeto envolve a realização de testes em campo, concentrando-se inicialmente em uma fazenda específica. Essa abordagem visa aprofundar a validação do protótipo em condições reais e aprimorar sua eficácia na coleta de dados de produção leiteira. A coleta de dados em campo é crucial para avaliar a adaptabilidade do protótipo às nuances específicas de cada fazenda, contribuindo para refinamentos e melhorias contínuas. Essa estratégia não apenas fortalecerá a confiabilidade do sistema, mas também fornecerá informações valiosas para futuras implementações em escala mais ampla.

A Tabela 10 apresenta dados relacionados à produção de leite ao longo de cinco dias na Fazenda, identificando as quantidades em quilogramas (kg) para cada animal, numerado de 1 a 13, durante cada dia específico.

Tabela 10 - Dados obtidos relacionados com as *tags* para a Fazenda

Tag	Dia 1(kg)	Dia 2(kg)	Dia 3(kg)	Dia 4(kg)	Dia 5(kg)
1	11,1	10,2	10,5	9,9	12,9
2	9,8	10,5	11,2	10,5	10,3
3	10,5	10,2	11,0	9,6	10,5
4	9,0	9,2	10	8,9	9,9
5	10	10,5	9,5	7,9	10,5
6	15,2	15,2	16,0	14,0	16,2
7	12,2	13,0	14,0	12,0	11,7
8	10,7	10,9	11,0	9,0	9,8
9	8,2	8,5	9,0	8,0	8,7
10	5,1	5,3	5,6	5,0	5,9
11	14,4	14,8	15,0	13,0	14,2
12	8,2	8,7	8,5	8,0	8,0
13	6,9	7,3	7,6	7,5	7,0

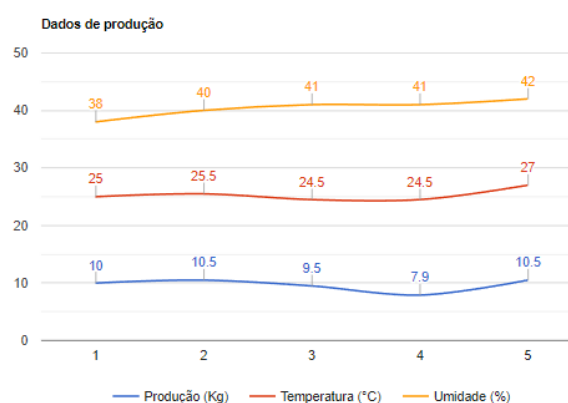
Fonte: Elaborado pelo autor (2022).

A análise dos dados pode revelar padrões de produção, variações diárias e desempenho individual dos animais. A interpretação desses números pode ajudar a identificar tendências,

avaliar a eficiência da ordenha, e fornecer *insights* sobre o desempenho geral e individual do rebanho em relação à produção de leite. Essa informação é vital para fazendeiros e pesquisadores que buscam otimizar a produção leiteira e otimizar seus lucros.

Para otimizar a apreciação dos dados, é possível examinar a Figura 47, obtida diretamente do sítio do protótipo:

Figura 47 - Gráfico de produção da vaca Nambuca



Fonte: Elaborado pelo autor (2022).

Os dados apresentados na Figura 45 correspondem a um único animal ao longo de cinco dias consecutivos de produção de leite. A variabilidade nas métricas, a temperatura ambiente em graus Celsius, a umidade percentual e a produção de leite em quilogramas, oferece uma análise detalhada do desempenho desse animal específico. A temperatura e a umidade capturam as condições ambientais que podem impactar diretamente o bem-estar do animal e, conseqüentemente, a produção de leite. A quantidade de leite produzida, expressa em quilogramas, destaca a contribuição individual desse animal para o volume total. A avaliação desses dados ao longo do tempo é crucial para identificar padrões, ajustar condições ambientais e aprimorar a gestão para maximizar a produção leiteira do animal, ou até a possibilidade do seu descarte.

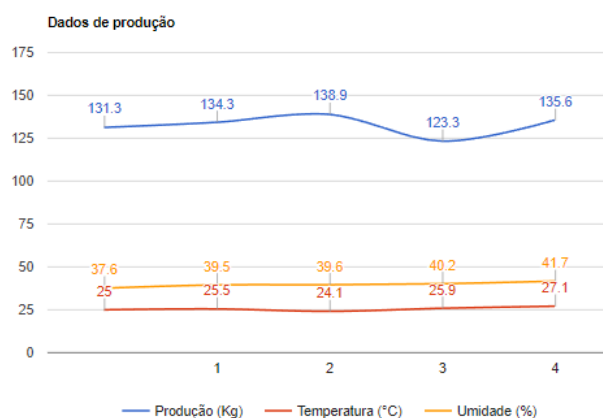
Figura 48 - Gráfico de produção total no período de 5 dias



Selecione o animal para gerar o gráfico:

Selecione ▾

Enviar



Projeto desenvolvido no IFMG - campus formiga.

Fonte: Elaborado pelo autor (2022).

Além da análise minuciosa do desempenho individual de um animal ao longo de cinco dias, é essencial contextualizar esses dados em relação às métricas gerais do rebanho, também disponíveis no protótipo. Enquanto as informações específicas de um único animal oferecem uma perspectiva micro, os dados agregados proporcionam uma visão macro da produção leiteira. O gráfico da Figura 46 apresenta as variações de temperatura, umidade e produção de leite para todo o rebanho, permitindo a identificação de padrões e correlações globais. Essa abordagem comparativa desempenha um papel crucial na implementação de estratégias

holísticas de manejo, visando otimizar a eficiência e a produção leiteira em escala, promovendo práticas sustentáveis e eficazes para o bem-estar do rebanho como um todo.

Figura 49 - Dados da produção da vaca Nambuca dispostos na forma tabular



Selecione o animal :

Selecione ▾

Enviar

Tabela de informações da Nambuca

Dia	Tempo (s)	Temperatura (°C)	Umidade (%)	Produção (Kg)
1	230	25	38	10
2	220	25.5	40	10.5
3	210	24.5	41	9.5
4	197	24.5	41	7.9
5	231	27	42	10.5

Projeto desenvolvido no IFMG - campus formiga.

Fonte: Elaborado pelo autor (2022).

Figura 50 - Dados da produção total dispostos em tabela



Tabela de informações da produção diária total

Dia	Tempo (s)	Temperatura (°C)	Umidade (%)	Produção (Kg)
1	209.92	25,0	37,6	131,3
2	206.85	25,5	39,5	134,3
3	187.46	24,1	39,6	138,9
4	186.08	25,9	40,2	123,3
5	201.54	27,1	41,7	135,6

Projeto desenvolvido no IFMG - campus formiga.

Fonte: Elaborado pelo autor (2022).

Juntamente com a representação visual por meio de gráficos, é fundamental também dispor esses dados em formato tabular, como evidenciado nas tabelas representadas nas Figuras 47 e 48. As tabelas oferecem uma apresentação estruturada e detalhada das métricas do desempenho individual e coletivo do rebanho, proporcionando uma referência direta para a análise. Complementando as informações visuais dos gráficos adicionando também o tempo de ordenha que reflete a efetividade operacional do processo, essas tabelas enriquecem a compreensão dos dados, permitindo uma interpretação mais aprofundada e precisa. Essa abordagem abrangente, que inclui tanto representações gráficas quanto tabelares, aprimora significativamente a capacidade de interpretação, facilitando a tomada de decisões fundamentadas para otimizar a produção leiteira no contexto do rebanho.

A implementação prática deste sistema em campo representa uma fase crucial voltada para a avaliação da confiabilidade do protótipo em ambientes reais, especialmente ao observar e analisar novos conjuntos de dados provenientes de outra fazenda. Este novo teste em campo destaca a importância de verificar a adaptabilidade e eficácia do sistema em contextos diversos, com ênfase na confiabilidade das operações. A observação atenta desses novos conjuntos de

dados não apenas enriquece a compreensão do desempenho do sistema em diferentes cenários, mas também contribui para fortalecer a confiança na aplicação prática do protótipo.

#### 4.1 Levantamento de custos

No decorrer da elaboração do protótipo, foram empregados um *Raspberry Pi Model B+*, um sensor DHT11, um módulo RFID MRFC522, quatro células de carga com capacidade de 20 kg cada e quatro módulos conversores HX711. Todos os dispositivos foram cotados em 01/11/2023, e os respectivos custos associados são detalhados na Tabela 11.

Tabela 11 - Levantamento de custos para o protótipo

Equipamento	Loja	Quantidade	Preço total
Raspberry pi model B+	Mercado livre	1	R\$ 698,97
DHT11	Arducore	1	R\$ 13,14
MRFC522	Amazon	1	R\$ 34,75
Chaveiro 13,56 MHz	Baú da eletrônica	10	R\$ 22,70
Célula de carga 20 kg (incluído conversor hx711)	Mercado livre	4	R\$ 100,00
Resistor 4K7 5% (1/4W)	Baú da eletrônica	1	R\$0,06
Placas e parafusos para construção da balança	-	1	R\$ 105,00
Gastos para instalação do protótipo	-	-	R\$ 50,00
Total			R\$ 1025,62

## 5 CONCLUSÃO

O desenvolvimento do protótipo para automação leiteira revelou-se um processo abrangente, dividido em duas fases distintas. Na primeira etapa, realizou-se uma análise individual dos componentes, objetivando compreender suas operações e estabelecer a integração com o *Raspberry Pi*. A segunda fase concentrou-se na interligação de todas as partes do projeto, culminando na implementação prática em campo.

Durante a fase inicial, os resultados foram promissores, com os equipamentos demonstrando desempenho alinhado às expectativas delineadas em seus manuais. Aprofundar o entendimento do modelo de operação do *Raspberry Pi* foi um ponto crucial, dado seu sistema baseado em *Linux (Raspbian)*, exigindo familiaridade com sistemas menos intuitivos em comparação aos convencionais da *Microsoft*.

A escolha da linguagem de programação *Python* revelou-se acertada, facilitando o desenvolvimento eficiente de códigos de teste. A configuração do *Raspberry Pi* como servidor *web* e a criação do site foram etapas conduzidas sem contratempos, enquanto a modificação do adaptador *Wi-Fi* para atuar como ponto de acesso representou o desafio mais substancial, exigindo atenção aos detalhes.

A avaliação da balança proporcionou resultados satisfatórios, evidenciando uma precisão e uma exatidão do desvio padrão abaixo da resolução da balança. Como esses resultados foram satisfatórios, a segunda parte do projeto foi implementada, envolvendo a coleta de dados em campo, essa coleta foi realizada em uma fazenda situada do município de Bambuí-MG.

A análise dos dados da fazenda 1 indicou variações na produção de leite ao longo dos dias, revelando padrões e possibilitando a avaliação do desempenho individual dos animais. A fazenda 2, por sua vez, apresentou maior homogeneidade na produção, destacando a importância de considerar as particularidades de cada rebanho.

Os gráficos gerados pelo protótipo proporcionaram uma visualização eficaz da produção em relação à temperatura, umidade e suas médias. Essas informações são cruciais para fazendeiros e pesquisadores, possibilitando otimizar a produção leiteira e compreender fatores ambientais influenciadores.

Além disso, o levantamento de custos revelou um investimento total de R\$ 1.025,62 para a implementação do protótipo. Esse conhecimento financeiro é essencial para futuras implementações e possibilita uma avaliação completa do projeto em termos de viabilidade econômica.

Em síntese, o protótipo não apenas demonstrou eficácia técnica, mas também destacou a relevância prática da automação leiteira em ambientes agrícolas. Com resultados positivos e custos de implementação claros, este projeto serve como um ponto de partida valioso para avanços futuros no campo da monitorização automatizada da produção leiteira.

## 6 TRABALHOS FUTUROS

Com base nas investigações realizadas neste estudo, algumas sugestões promissoras para trabalhos futuros incluem:

- Integrar o sistema para possibilitar o monitoramento online da produção leiteira, viabilizando o acesso em tempo real às informações pelos proprietários, inclusive quando estão fora da propriedade.
- Criar um sistema de notificação que informe os proprietários acerca de eventos críticos, tais como falhas nos equipamentos ou variações abruptas de produção.
- Avaliar a incorporação de sensores mais avançados para medição de temperatura e umidade, buscando integrá-los a equipamentos que permitam o controle efetivo desses fatores.
- Implementar um sistema de cálculo para determinar a quantidade de concentrado a ser fornecido a cada animal, utilizando os dados do projeto e considerando a relação custo-benefício entre a ração e produção.
- Atualizar o sistema operacional da versão *Stretch* para a mais recente versão *Buster*, uma vez que a primeira deixou de receber atualizações. Essa transição é essencial para obter os últimos recursos, correções de segurança e melhorias oferecidos pela *Raspberry Pi Foundation*. Além da atualização do sistema, é imperativo também atualizar as bibliotecas utilizadas nos projetos, assegurando compatibilidade e otimização de desempenho. Ao adotar a versão *Buster*, usuários e desenvolvedores poderão explorar todo o potencial das mais recentes inovações e aprimoramentos disponíveis no ecossistema *Raspberry Pi*, contribuindo para a continuidade e excelência dos seus projetos.

## REFERÊNCIAS BIBLIOGRÁFICAS

- 20KG., VOLTRIZ. S. D. P. – C. D. C. –. Disponível em:  
<https://www.voltriz.com.br/produto/sensor-de-peso-celula-de-carga-20kg/#tab-specification>.  
 Acesso em: 2021 fev. 15.
- B.V., NXP. MFRC522 Datasheet. **alldatasheet**, 2007. Disponível em:  
<https://www.alldatasheet.com/datasheet-pdf/pdf/227839/NXP/MFRC522.html>. Acesso em: 15 fev. 2021.
- BEGA, Egídio A. **Instrumentação Industrial**. 3. ed. São Paulo: Interciência, 2011.
- BORGES, Luis E. **PYTHON PARA DESENVOLVEDORES**. São Paulo: Novatec, 2014.
- BORGES, Luiz E. **PYTHON PARA DESENVOLVEDORES**. São Paulo: Novatec, 2014.
- COMER, Douglas E.. **Redes de computadores e internet**. 6ª. ed. São Paulo: Bookman, 2016.
- CONVERSE, Tim; PARK, Joyce. **PHP a bíblia**. 2º. ed. Rio de Janeiro: Elsevier, 2002.
- CROTTI, Y. *et al.* Raspberry Pi e Experimentação Remota, 2021. Disponível em:  
[http://www.icbl-conference.org/proceedings/2013/papers/Contribution49\\_a.pdf](http://www.icbl-conference.org/proceedings/2013/papers/Contribution49_a.pdf). Acesso em: 05 fev. 2021.
- ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. São Paulo: Pearson, 2005.
- EMBRAPA. Embrapa. **Controle leiteiro reduzido para propriedades leiteiras de economia familiar: características da tecnologia.**, 2016. Disponível em:  
<https://www.embrapa.br/busca-de-publicacoes/-/publicacao/1060088/control-leiteiro-reduzido-para-propriedades-leiteiras-de-economia-familiar-caracteristicas-da-tecnologia>.  
 Acesso em: 01 abr. 2023.
- EMBRAPA. Cadeia produtiva do leite no Brasil. **Embrapa**, 01 Abr. 2020. Disponível em:  
<https://ainfo.cnptia.embrapa.br/digital/bitstream/item/215880/1/CT-123.pdf>. Acesso em: 01 abr. 2023.
- EMBRAPA. ANUÁRIO Leite 2022: pecuária leiteira de precisão. **Embrapa**, 2022.  
 Disponível em: <https://www.embrapa.br/busca-de-publicacoes/-/publicacao/1144110/anuario-leite-2022-pecuaria-leiteira-de-precisao>. Acesso em: 01 abr. 2023.
- FOUNDATION, Raspberry P. raspberry-pi-3-model-b-plus. **raspberrypi**, 2023. Disponível em: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>. Acesso em: 01 fev. 2023.
- GUS. How to setup a Raspberry Pi RFID RC522 Chip. **pimylifeup**. Disponível em:  
<https://pimylifeup.com/raspberry-pi-rfid-rc522/>. Acesso em: 05 fev. 2021.
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. PPM - Pesquisa da Pecuária Municipal, 2023. Disponível em:  
<https://www.ibge.gov.br/estatisticas/economicas/agricultura-e-pecuaria/9107-producao-da-pecuaria-municipal.html?=&t=destaques>. Acesso em: 27 Abr. 2023.
- JUNIOR, Joel A. RFID – identificação por radiofrequência. **Monografias**, 2007. Disponível em: <https://www.monografias.com/pt/trabalhos3/rfid-identificacao-radiofrequencia/rfid-identificacao-radiofrequencia3.shtml>. Acesso em: 03 jul. 2023.

- JUNIOR, Sergio L. B. Medição de nível. **IFSC**, 2017. Disponível em: <http://docente.ifsc.edu.br/sergio.brockveld/MaterialDidatico/Instrumenta%C3%A7%C3%A3o/Aula%20-%20Medi%C3%A7%C3%A3o%20de%20N%C3%ADvel.pdf>. Acesso em: 14 JUN. 2023.
- LOPES, João G.; PESTRE, João ; TORRES , Matheus. Identificação por Rádiofrequência. **ufrj**, 2018. Disponível em: <https://www.gta.ufrj.br/ensino/eel878/redes1-2018-1/trabalhos-vf/rfid/introducao.html>. Acesso em: 14 jun. 2023.
- MELO, Daniel T. D. **Apostila de introdução ao CSS**. São Paulo: Mococa, 2015.
- MENEZES, Nilo N. C. **ntrodução à Programação com Python**: algoritmos e lógica de programação para iniciantes. São Paulo: Novatec, 2010.
- MILKPOINT. Fazenda Colorado: divisão de lotes por produção de leite garante maior eficiência. **MilkPoint**, 01 Abr. 2023. Disponível em: <https://www.milkpoint.com.br/colunas/educapoint/um-tour-pelos-setores-da-maior-produtora-de-leite-do-pais-104784n.aspx>.
- NIEDERAUER, Juliano. **Desenvolvendo Websites com PHP**. 3ª. ed. São Paulo: Novatec, 2017.
- PORTAL CELULA DE CARGA. Strain Gages. **celuladecarga**, 2023. Disponível em: <http://celuladecarga.com.br/17/strain-gages-extensometros-eletricos/>. Acesso em: 14 jun. 2023.
- PREDIGER, Daniel ; SILVEIRA, Sidnei Renato ; FREITAS, Edison Pignaton de. Revistas de sistemas e computação - RSC. **unifacs**, 2014. Disponível em: <https://revistas.unifacs.br/index.php/rsc/article/view/3620>. Acesso em: 04 jul. 2023.
- REZENDE, Ricardo. Conceitos Fundamentais de Banco de Dados. **devmedia**, 2023. Disponível em: <https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>. Acesso em: 04 jul. 2023.
- SEMICONDUCTOR, Avia. 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales. **Alldatasheet**, 2023. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/pdf/1132222/AVIA/HX711.html>. Acesso em: 12 jul. 2023.
- SENAI. intrumentação. **ifrn**, 1999. Disponível em: <https://docente.ifrn.edu.br/gustavosouza/2012.2/3-qui-int-1v/Instrumentacao%20Basica%20I%20-%20Pressao%20e%20Nivel%20-%20SENAI.pdf>. Acesso em: 13 jun. 2023.
- SILVA, Mauricio S. **Criando sites com HTML**: sites de alta qualidade com html e css. São Paulo: Novatec, 2014.
- SILVA, Prof. J. H. D. D. Precisão, Exatidão, e a Terminologia das Medições. **unesp**, 2006. Disponível em: [http://wwwp.fc.unesp.br/~jhdsilva/Precisao\\_e\\_Terminologia\\_de\\_Medicoes](http://wwwp.fc.unesp.br/~jhdsilva/Precisao_e_Terminologia_de_Medicoes). Acesso em: 2023 jul. 04.
- SUNROM TECHNOLOGIES. **DHT11**: humidity and temperature sensor., 2021. Disponível em: <https://datasheetspdf.com/pdf/785592/Sunrom/DHT11/1>. Acesso em: 16 fev. 2021.
- THOMAZINI, Daniel; ALBUQUERQUE, Pedro Urbano Braga de. **Sensores industriais**: fundamentos e aplicações. 9ª. ed. São Paulo: Érica, 2020.
- THOMSEN, Adilson. Monitorando Temperatura com DHT11 e Raspberry Pi. **makerhero**, 2015. Disponível em: <https://www.makerhero.com/blog/temperatura-umidade-dht11-com-raspberry-pi/>. Acesso em: 20 fev. 2021.

TOTVS. O que é HTML. **TOTVS**, 2023. Disponível em:

<https://www.totvs.com/blog/developers/o-que-e-html/>. Acesso em: 04 jul. 2023.

UFRJ. RFID - Como funciona. **ufrj**, 2023. Disponível em:

[https://www.gta.ufrj.br/grad/07\\_1/rfid/RFID\\_arquivos/o%20que%20e.htm](https://www.gta.ufrj.br/grad/07_1/rfid/RFID_arquivos/o%20que%20e.htm). Acesso em: 04 jul. 2023.

UFRN. Medidor de nível por pesagem. **Instituto Metrópole Digital**, 2013. Disponível em:

<https://materialpublic.imd.ufrn.br/curso/disciplina/1/52/8/4>. Acesso em: 04 jul. 2023.

WEBER, Albano L. ufpr. **Metodologia e Instrumentação**, 2008. Disponível em:

[http://servidor.demec.ufpr.br/disciplinas/TM117/Arquivos%20H%E9lio/Strain\\_gauge%20extensometria.pdf](http://servidor.demec.ufpr.br/disciplinas/TM117/Arquivos%20H%E9lio/Strain_gauge%20extensometria.pdf). Acesso em: 01 jul 2023.

WENDLING, Marcelo. Sensores. **UNESP**, 2010. Disponível em:

<https://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>. Acesso em: 14 jun 2023.

## ANEXO A

ler.py ✕

```
1 from time import sleep
2 import sys
3 from mfrc522 import SimpleMFRC522
4 reader = SimpleMFRC522()
5
6 try:
7     while True:
8         print("Segure a 'tag' perto do sensor:")
9         id, text = reader.read()
10        print("ID: %s\nTexto: %s" % (id,text))
11        sleep(5)
12 except KeyboardInterrupt:
13     GPIO.cleanup()
14     raise
```

escrever.py ✕

```
1 #!/usr/bin/env python
2
3 import RPi.GPIO as GPIO
4 import SimpleMFRC522
5
6 reader = SimpleMFRC522.SimpleMFRC522()
7 try:
8     while True:
9         text = raw_input('New data:')
10        print("Now place your tag to write")
11        reader.write(text)
12        print("Written")
13 except KeyboardInterrupt:
14     GPIO.cleanup()
15     raise
16 |
```

## ANEXO B

DHT11.py ✕

```
1
2 # Carrega as bibliotecas
3 import Adafruit_DHT
4 import RPi.GPIO as GPIO
5 import time
6
7 # Define o tipo de sensor
8 sensor = Adafruit_DHT.DHT11
9
10
11 GPIO.setmode(GPIO.BOARD)
12
13 # Define a GPIO conectada ao pino de dados do sensor
14 pino_sensor = 24
15
16 def TempHum():
17     umid, temp = Adafruit_DHT.read_retry(sensor, pino_sensor);
18     print (temp,umid)
19     return (umid,temp)
20 TempHum()
```

## ANEXO C

## Coleta.py

```

1  # Define a localização do interpretador Python
2  # Importa a biblioteca para controle dos pinos GPIO
3  from RPi.GPIO import as GPIO
4  # Importa a biblioteca para comunicação com o leitor RFID
5  from mfrc522 import SimpleMFRC522
6  # Importa a biblioteca para manipulação de tempo
7  import time
8  # Importa a biblioteca para interação com o sistema operacional
9  import os
10 # Importa a biblioteca para comunicação com o sensor de umidade e temperatura DHT11
11 import Adafruit_DHT
12 # Importa a biblioteca para comunicação Python-MySQL
13 import mysql.connector
14
15 # Parâmetros para configuração da conexão com o banco de dados
16 db = mysql.connector.connect(user='pi', password='raspberry',
17                             host='localhost',
18                             database='Fazenda')
19 cursor = db.cursor()
20
21 # Função para obter o peso
22 def Peso():
23     global val # Variável para receber peso
24     GPIO.cleanup() # Limpa o modo de configuração dos GPIO
25     from hx711 import HX711
26
27     try:
28         # Configuração do sensor de peso HX711
29         hx711 = HX711(
30             dout_pin=5,
31             pd_sck_pin=6,
32
33             channel='A',
34             gain=64
35         )
36
37         hx711.reset() # Antes de começar, reinicia o HX711 (não obrigatório)
38         measures = hx711.get_raw_data(3)#Realiza 3 medidas de peso para maior certeza
39     finally:
40         GPIO.cleanup() # Sempre faça uma limpeza dos pinos GPIO nos scripts!
41         referencia = -16000 #Valor de referencia da balança
42         tara = 39000 * 3 #Valor da tara da balança * quantidade de medidas realizada
43         val = ((sum(measures) + tara) / len(measures)) / referencia
44         val = round(val, 1)#arredonda o valor para 1 casa decimal
45
46 # Função para obter dados do sensor de umidade e temperatura DHT11
47 def Sensor():
48     global umidade
49     global temperatura
50     sensor = Adafruit_DHT.DHT11
51     GPIO.setmode(GPIO.BOARD)
52     pino_sensor = 24 #define o pino para obter dados do DHT11
53     umidade, temperatura = Adafruit_DHT.read_retry(sensor, pino_sensor)
54
55 # Função para enviar dados para o servidor MySQL
56 def sendDataToServer():
57     global temperatura
58     global nivel
59     global umidade
60     global nome
61     global tempo
62     global val

```

```

62     from datetime import date
63
64     data_atual = date.today()
65     Sensor()
66     temp = "%.1f" % temperatura
67     hum = "%.1f" % umidade
68     nivel = '%.2f' % peso
69     tem = "%.1f" % tempo# Converte as variaveis para o farmato aceitavel no banco  ↵
    de dados
70     cursor.execute("INSERT INTO Tabela (Tempo, Temperatura, Umidade, Producao, Nome) \
71                   VALUES (%s,%s,%s,%s,%s)", (tem, temp, hum, nivel, nome))# envia  ↵
    os valores para o banco de dados
72     db.commit()
73     reader = SimpleMFRC522()
74
75     # Inicializa variáveis globais
76     aux = 1
77     global nome
78     global tempo
79     global peso
80     global val
81
82     # Loop principal do programa que executa até a aproximação de qualquer tag
83     while True:
84         if aux:
85             id, text = reader.read()
86             aux = 0
87             aux2 = text
88             print("Inicio")
89             start = time.time()
90             Peso()
91             pesoi = val
92             print(text) # identificacao do inicio da obtenção de dados do animal  ↵
    identificado na tag
93
94             time.sleep(3)
95             if (aux == 0):
96                 id, text = reader.read()
97                 if (aux2 == text):
98                     aux = 1
99                     print("Fim")
100                    print(text)
101                    stop = time.time()
102                    Peso()
103                    pesof = val
104                    peso = pesof - pesoi
105                    tempo = stop - start
106                    nome = text
107                    sendDataToServer()
108                    time.sleep(3)
109
110     # Limpa os pinos GPIO e encerra o programa
111     GPIO.cleanup()
112     print('Programa encerrado.')

```