



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

JOÃO VICTOR DE BARROS FELIX

**Estudo de desenvolvimento de sistema embarcado para monitoramento de
parâmetros gasosos em sistemas de confinamento de gado de leite**

RECIFE

2023

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

JOÃO VICTOR DE BARROS FELIX

Estudo de desenvolvimento de sistema embarcado para monitoramento de parâmetros gasosos em sistemas de confinamento de gado de leite

TCC apresentado ao Curso de Bacharelado em Engenharia da Computação da Universidade Federal de Pernambuco, Centro DE INFORMÁTICA, como requisito para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador(a): Adriano Augusto de Moraes Sarmiento
Coorientador(a):

RECIFE

2023

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Barros Felix, João Victor de.

Estudo de desenvolvimento de sistema embarcado para monitoramento de parâmetros gasosos em sistemas de confinamento de gado de leite / João Victor de Barros Felix. - Recife, 2023.

32 p

Orientador(a): Adriano Augusto de Moraes Sarmiento

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Engenharia da Computação - Bacharelado, 2023.

1. Sistemas embarcados. 2. Monitoramento gasoso. 3. Pecuária Leiteira. 4. Confinamento bovino. I. Moraes Sarmiento, Adriano Augusto de. (Orientação).
II. Título.

000 CDD (22.ed.)

RESUMO

Este trabalho teve como objetivo o desenvolvimento de um sistema embarcado para o monitoramento de amônia e metano em sistemas de confinamento bovinos. O sistema utiliza uma arquitetura em fog, com dois nós coletores das informações, implementados em módulos ESP32, que se comunicam através de LoRaMESH com um terceiro dispositivo, composto de uma Raspberry Pi com Arduino para processar tais dados e disponibilizá-los em um sistema de Business Intelligence (BI) online. O sistema de BI cumpre a função de gerar informações relevantes para o usuário e facilitar a sua visualização. Ao fim da execução foi obtido um sistema em pleno funcionamento e capaz de indicar tendências sobre os dados, detectando variações de rotinas no confinamento e impacto destes nos níveis monitorados, porém ainda apresentando limitações e espaço para melhorias futuras.

Palavras-chave: Sistemas embarcados; Monitoramento gasoso; Pecuária Leiteira; Confinamento bovino.

ABSTRACT

The goal of this project was the development of an embedded system for monitoring ammonia and methane in bovine confinement systems, using a fog architecture, with two information collecting nodes, implemented in ESP32 modules and using LoRaMESH communication with a third device, using a Raspberry Pi with Arduino to process such data and make it available in an online Business Intelligence system. The BI system fulfills the function of generating relevant information for the user and facilitating its visualization. In the end, it was obtained a fully functioning system capable of indicating trends in the data, detecting variations in routines in confinement and their impact on the monitored levels, but still presenting problems and room for future improvements.

Keywords: Embedded systems; Gas monitoring; Dairy Farming; Cattle confinement;

SUMÁRIO

INTRODUÇÃO	6
CONTEXTO	6
MOTIVAÇÃO E JUSTIFICATIVA	7
OBJETIVO	7
ESTRUTURA DA MONOGRAFIA	7
REFERENCIAL TEÓRICO	8
LORA	8
LORAMESH	8
METABASE	10
SISTEMA IOT DE MONITORAMENTO DE GASES	12
MÉTODO	12
ARQUITETURA DO SISTEMA	12
IMPLEMENTAÇÃO DO SISTEMA	13
CÓDIGO IMPLEMENTADO	18
AMBIENTE DE TESTES	24
RESULTADOS	27
CONCLUSÕES E TRABALHOS FUTUROS	30
REFERÊNCIAS	31

1. INTRODUÇÃO

1.1 Contexto:

A produção de leite no Brasil vem apresentando crescimento nos últimos anos, tanto em termos de produção bruta, quanto em termos de valor agregado (valor por litro). Adicionalmente, há boas perspectivas de crescer ainda mais nos próximos anos, passando por uma expansão de produção e consumo, possuindo uma projeção de passar de uma produção de 34,1 bilhões de litros no ano de 2023 para 40,5 bilhões ao final de 2033, segundo o Ministério da Agricultura [1][2][6].

Neste contexto se encontram desafios para o meio, onde para acompanhar o crescimento, devem-se ter investimentos focados no aumento da produtividade e gestão do meio. Passando principalmente pelo pilar de acesso e adoção de tecnologias, onde na tendência atual se encontra um aumento dos custos de produção do meio, e em decorrência deste, apresentando uma diminuição na participação de pequenos e médios produtores.

Um outro ponto importante para os próximos anos são as metas para a redução dos impactos ambientais. O plano ABC+ do Governo Federal [3][4], visa reduzir a produção de carbono e possui um dos focos de atuação em sistemas de confinamento e semi-confinamento. Mudanças na alimentação e manejo dos animais permitem reduzir a emissão de tais gases.

Uma tendência observada pelo Centro de Estudos Avançados em Economia Aplicada (CEPEA) [8] da Universidade de São Paulo nos últimos anos é uma migração dos sistemas de produção nas propriedades para o confinamento, um investimento por parte dos produtores que visa o ganho em escala de produção. Por sua vez, gerando também um aumento nos custos de produção, mas que tende a compensar de forma proporcional por um aumento nas receitas.

A tendência no confinamento bovino [7], entretanto, acaba gerando novos riscos, onde o acúmulo de metano decorrentes da produção já ocasionou episódios de explosões. Além disto, os efeitos do gás amônia e seus óxidos se mostram muito corrosivos [9][10] a elementos estruturais de concreto armado, bem como estruturas metálicas [11] e implementos utilizados para o manejo, oferecendo riscos e uma redução da vida útil dos mesmos caso este não seja bem controlado.

1.2 Motivação e Justificativa:

Diante do exposto, este estudo visa analisar o desenvolvimento de uma solução de baixo custo em Internet das coisas (IOT) que viabilize o monitoramento de tendências na emissão de metano e amônia em sistemas de confinamento de gado leiteiro.

Nesse estudo, barreiras de implementação e adoção de tal solução serão analisadas, com o intuito de facilitar o acesso a tecnologias que melhorem a gestão no meio. Com isso espera-se mitigar riscos à integridade de elementos estruturais, atuar na redução da emissão de carbono e prover um melhor aproveitamento dos recursos utilizados, como implementos agrícolas, estes compostos por materiais corrosivos e estando em contato recorrente com este meio.

1.3 Objetivo:

Este trabalho visa disponibilizar um sistema de auxílio na gestão e tomada de decisão sobre alimentação e manejo em sistemas de confinamento. Tornando perceptível se mudanças nestas impactaram de forma positiva ou negativa no controle da emissão e acúmulo dos gases metano e amônia.

Os objetivos específicos em relação a esse trabalho são definidos em:

1. Desenvolver um sistema IOT para monitoramento dos níveis dos gases Metano e Amônia;
2. Testar o sistema em um ambiente real;
3. Analisar o comportamento do sistema e possíveis barreiras em relação a sua adoção.

1.4 Estrutura da Monografia:

A divisão deste trabalho será abordada da seguinte forma: o Capítulo 2 traz o referencial teórico necessário para o entendimento das tecnologias utilizadas. O Capítulo 3 descreve quais os materiais utilizados, a metodologia e a implementação empregadas para a realização dos experimentos. O Capítulo 4 descreve e analisa os resultados obtidos, indicando possíveis problemas e acertos do modelo. O Capítulo 5 apresenta as conclusões do trabalho, avaliando o que foi atingido e melhorias para versões futuras.

2. REFERENCIAL TEÓRICO

Este capítulo trata sobre os aspectos técnicos das tecnologias utilizadas, explicando mais sobre elas e indicando o seu papel na execução.

2.1: LoRa:

A comunicação LoRa é uma tecnologia de radiofrequência com foco em alto alcance, sendo a sigla desta vinda de *Long Range* (Longo Alcance em português). Esta rede mantém comunicações em uma baixa frequência quando comparado a outras redes, possuindo homologação no Brasil para atuação em 915MHz, distante da faixa convencional de 2,4GHz.

Para manter uma comunicação confiável em altas distâncias, a rede LoRa acaba por adotar frequências mais baixas de operação, uma menor banda de operação e um sistema de modulação baseado em *Chirp Spread Spectrum* [17][18]. Esta modulação se baseia na codificação por incremento linear da frequência, onde todo o canal é utilizado de forma simultânea para a transmissão do mesmo sinal. Baseando a codificação no domínio do tempo, esta opção gera uma maior confiabilidade na transmissão em longas distâncias.

2.2: LoRaMESH:

A rede mais comum que utiliza o LoRa é a rede LoRaWAN, onde consiste em um gateway central, responsável por realizar a comunicação com os demais nós da rede, possuindo a comunicação com a interface de Internet. A LoRaWAN, entretanto, traz problemas em alguns cenários onde mesmo sendo uma rede focada em alto alcance, esbarra em uma limitação de alcance, ficando restrito ao alcance do Gateway. Assim, para ampliar o alcance, foi proposto uma rede Mesh de LoRa, o LoRaMESH, baseado em múltiplos saltos entre nós.

A rede LoRaMESH foca em ser mais descentralizada, onde os nós participam da comunicação para ampliar o alcance, retransmitindo pacotes curtos entre si, e fazendo com que a limitação do alcance da rede não seja mais apenas o alcance do gateway. Esta abordagem traz a clara vantagem de ampliar a possibilidade de escalabilidade para a rede e também a aplicação, porém acaba impactando em um

maior consumo energético por meio dos dispositivos, uma vez que estes ficam também responsáveis pela retransmissão.

Os módulos LoRaMESH da Radioenge permitem a criação de uma rede onde o dispositivo com identificador 0 será o nó principal, na rede, o responsável por receber o tráfego final de todos os módulos, e onde os demais módulos podem ser configurados em duas classes. Com a configuração em classe A, o módulo transmissor é ativado apenas durante a transmissão do seu pacote, e uma janela de tempo subsequente para recepção, que pode ser configurada com duração de 5, 10 ou 15 segundos.

Para a configuração em classe C, o módulo passa a atuar com o papel de repetidor na rede, onde este permanece sempre ligado para recepção, podendo receber pacotes de outros módulos que não possuem acesso a pontos da rede para que estes sejam retransmitidos. Para a classe C, o modo de transmissão é ativado apenas quando necessário transmitir um novo pacote. Esta configuração implica em um maior alcance da rede, uma vez que esta não depende apenas do alcance do módulo 0, entretanto implica em um maior consumo energético, devido ao fato de permanecer ligado para recepção.

Também é possível configurar os módulos em relação a largura de banda utilizada, com as opções de 125, 250 ou 500 kHz, ao Spreading Factor, com opções de 7, 8, 9, 10, 11 ou 12 e com a configuração de Coding Rate, com opções de 4/8, 4/7, 4/6 e 4/5, e cujo conjunto de todos estes fatores implica em uma diferença de alcance e taxa de transmissão, como pode ser indicado pela tabela presente no datasheet dos módulos, descritas na Figura 1.

Figura 1 - Tabelas de taxa de dados dos módulos LoRaMESH

Tabela 5: Taxa de dados LoRa - Coding Rate 4/8

Taxa de dados (bps)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	3418	1953	1098	610	336	183
	250	6836	3906	2197	1220	671	366
	500	13672	7813	4395	2441	1343	732
Coding Rate 4/8							

Tabela 6: Taxa de dados LoRa - Coding Rate 4/7

Taxa de dados (bps)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	3906	2232	1256	698	384	209
	250	7813	4464	2511	1395	767	419
	500	15625	8929	5022	2790	1535	837
Coding Rate 4/7							

Tabela 7: Taxa de dados LoRa - Coding Rate 4/6

Taxa de dados (bps)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	4557	2604	1465	814	448	244
	250	9115	5208	2930	1628	895	488
	500	18229	10417	5859	3255	1790	977
Coding Rate 4/6							

Tabela 8: Taxa de dados LoRa - Coding Rate 4/5

Taxa de dados (bps)		Spreading Factor					
		7	8	9	10	11	12
Largura de Banda (kHz)	125	5469	3125	1758	977	537	293
	250	10938	6250	3516	1953	1074	586
	500	21875	12500	7031	3906	2148	1172
Coding Rate 4/5							

Fonte: Datasheet Radioenge [19], 2021.

2.3 Metabase:

O sistema de BI (business intelligence) Metabase é uma ferramenta desenvolvida na linguagem Java e mantida por meio de código aberto. Devido a este fato, pode ser implementado sem o custo de um sistema de licenciamento, restando apenas o custo com infraestrutura por parte das aplicações.

O principal papel deste sistema é realizar consultas em banco de dados sem a necessidade do conhecimento de aspectos técnicos destas consultas por parte do usuário, realizando a exposição dos resultados e tratamentos destes dados, bem

como a sua disponibilização para compartilhamento, facilitando a interação com os dados e reduzindo a barreira de entrada para tal meio.

3. SISTEMA IOT DE MONITORAMENTO DE GASES

3.1 Método:

O desenvolvimento da aplicação se deu por meio de uma sistema IOT em arquitetura de Fog Computing. Esse sistema conta com 2 nós responsáveis por coletar os dados referentes ao nível dos gases, e por um processador responsável por receber os dados dos coletores por uma rede LoRaMESH, processar e agregar os mesmos, e disponibilizar informações baseadas nessas em um sistema de Business Intelligence (BI) de forma online em um servidor remoto e offline por meio de um sistema hospedado no próprio dispositivo do processador.

Após a implementação do sistema e a coleta dos dados, foram analisados os parâmetros referentes a desempenho do sistema, considerando a performance de processamento, comunicação e consumo de energia, levando em consideração também variações dos cenários com o intuito de analisar o comportamento deste em um sistema de maior evolução.

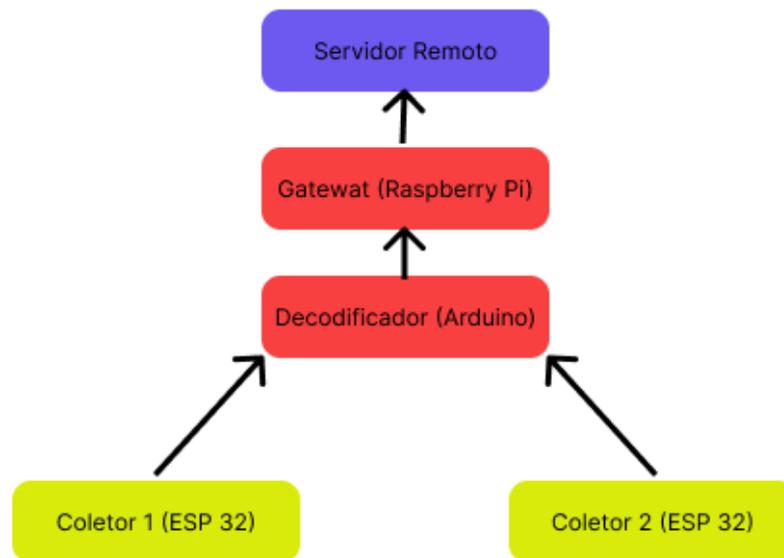
3.2 Arquitetura do Sistema:

A arquitetura do sistema foi adotada de uma forma a ser escalável e viabilizar a centralização dos dados coletados, e pode ser visualizada na Figura 2. Desta forma, apenas um dispositivo fica responsável por receber e decodificar a comunicação. O gateway fica responsável por fazer um tratamento inicial nos dados enviados pelos nós, sendo este o único responsável por possuir ligação da rede LoRa com a camada online.

Após a disponibilização dos dados ao banco de dados remoto, o usuário pode ter acesso a todo o tratamento e relacionamento destes, deixando esta atividade independente do processamento dos demais dispositivos. Porém, mantém-se a possibilidade de utilizar o poder de processamento da camada do gateway, uma vez que a Raspberry também recebe os dados disponibilizados em um banco executado em ambiente local.

Com esta arquitetura, ao ser necessário ampliar a rede em número de pontos de coleta, basta criar um novo nó, com o mesmo código e configurar o módulo LoRaMESH com um novo identificador.

Figura 2 - Arquitetura do Sistema



Fonte: O autor, 2023.

3.3 Implementação do Sistema:

Para a implementação do sistema foram utilizados dispositivos distintos, com seus respectivos objetivos, e sendo a placa ESP32 responsável por gerenciar a carga das baterias, uma vez que esta possui a estrutura necessária para tal.

Tabela de materiais utilizados:

Dispositivo	Quantidade	Valor unitário (R\$)
Placas WiFi LoRa ESP32 - HTIT-WB32LA	2	160,00
Módulos MQ-4	2	20,00
Módulos MQ-135	2	20,00
Placa Raspberry Pi 4	1	700,00
Arduino Pro mini	1	200,00
Módulos LoRaMESH Radioenge	3	200,00
Protoboard	3	12,00
Baterias de 3.7V - 650 mAh	2	50,00

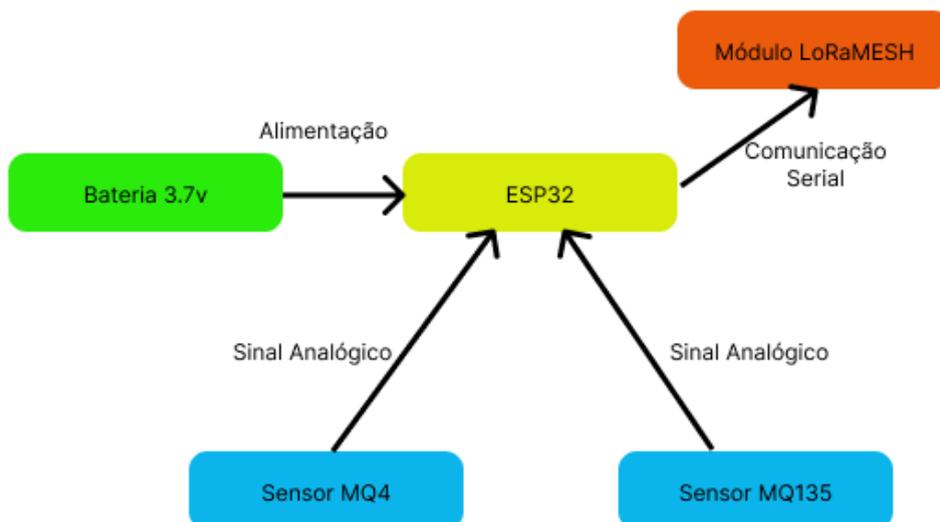
O custo aproximado de todo o equipamento ficou em torno de R\$1900,00. Sendo composto conforme a distribuição indicada na tabela.

Os custos com a camada online utilizada são disponibilizados em Dólar americano pelo fornecedor, ficando em U\$0,010 por hora de utilização do servidor, atingindo um teto máximo de cobrança de U\$7,00 por mês. O custo do banco de dados fica em U\$0,012 por hora de utilização do processamento, atingindo um teto máximo de cobrança de U\$9,00 por mês.

Os dispositivos responsáveis pela mensuração dos gases foram implementados nas placas ESP32, com o nível dos gases coletados a cada 1 minuto, armazenados temporariamente na placa, sendo feita a montagem e envio do pacote LoRa para o dispositivo gateway.

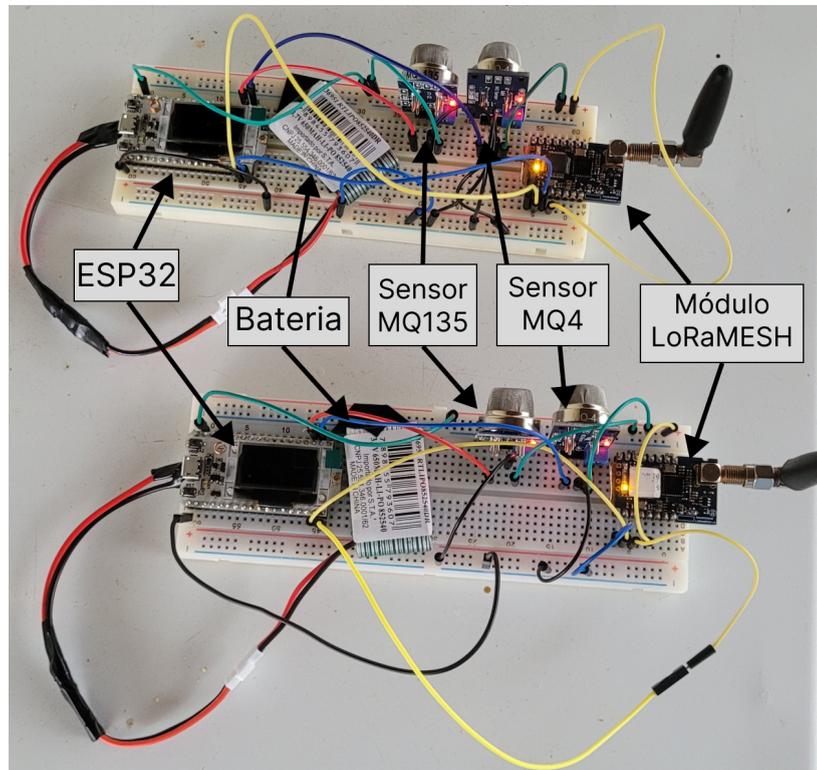
O diagrama dos dispositivos pode ser encontrado na Figura 3, onde a ESP recebe alimentação da bateria, os dados dos analógicos dos sensores, enviando mesmos em seus pacotes por comunicação serial com o módulo LoRaMESH. e os mesmos montados e prontos para instalação se encontram na Figura 4.

Figura 3 - Diagrama dos dispositivos coletores



Fonte: O autor, 2023.

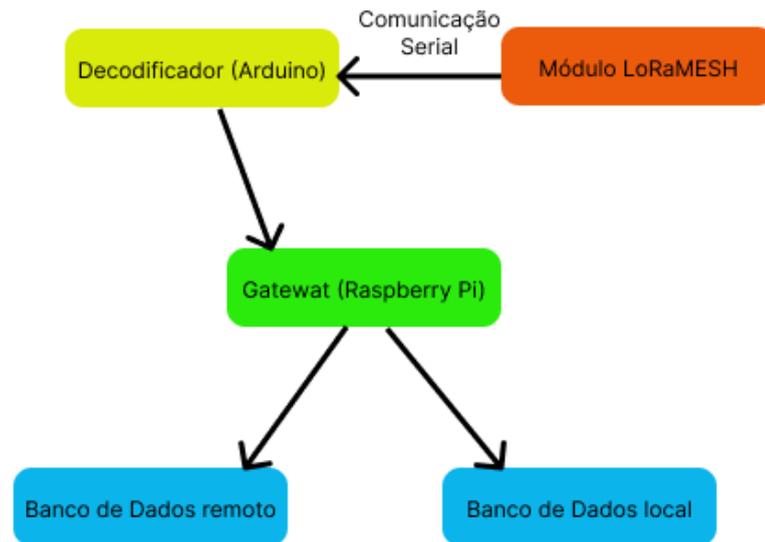
Figura 4 - Dispositivos nós responsáveis pela mensuração e transmissão dos dados



Fonte: O autor, 2023.

O dispositivo responsável por executar o papel de gateway e processar os dados foi montado em duas placas constituídas por um Arduino Pro mini e uma Raspberry Pi 4. O diagrama indicando a forma de comunicação destes dispositivos pode ser encontrado na Figura 5. A Arduino se encontra diretamente ligada ao módulo LoRaMESH, ficando responsável por receber e decodificar mensagem, enviando por comunicação serial via cabo USB para a Raspberry Pi. A mensagem completa recebida e o payload já decodificado para string, e contendo apenas as informações relevantes para a aplicação.

Figura 5 - Diagrama dos dispositivos do gateway



Fonte: O autor, 2023.

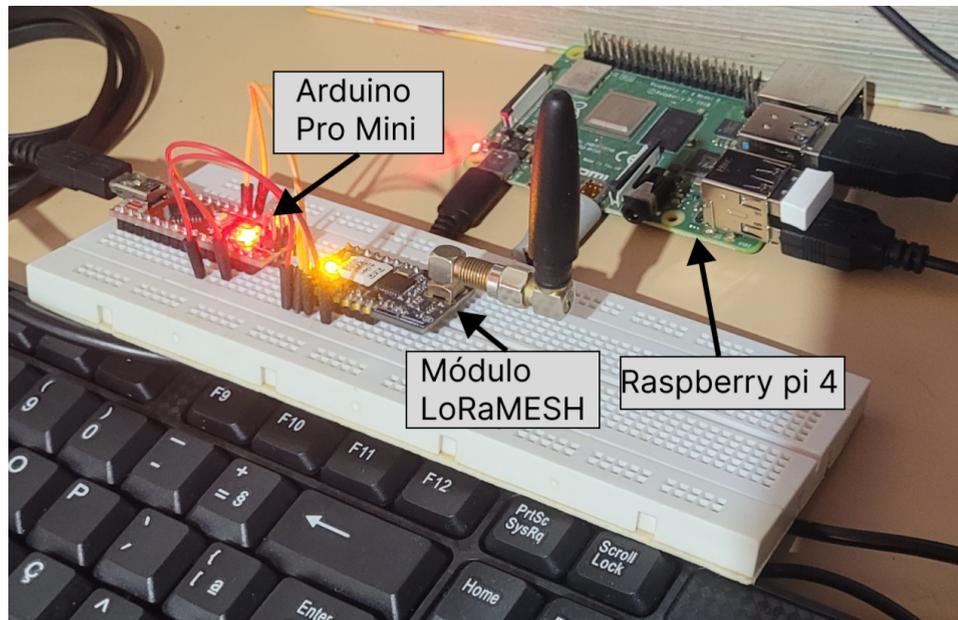
A Raspberry Pi por sua vez, ao receber as mensagens decodificadas pela Arduino, separa cada mensagem referente às 5 medições em tuplas de números inteiros, identifica o horário ao qual a mensagem foi recebida. Partindo do horário de recebimento são calculados os horários aproximados de cada uma das medições, calcula uma conversão de cada um dos valores dos gases do nível do sensor para o valor em partes por milhão (ppm), e insere o dado nos dois bancos de dados, o local e o remoto, responsáveis por armazenar e exibir os dados na plataforma de BI.

A escolha de se utilizar as duas placas conectadas se deu pelos motivos da biblioteca oficial da Radioenge para os módulos LoRaMESH estar disponível apenas para a plataforma arduino [16]. Para a utilização na Raspberry Pi necessitaria de uma reimplementação de C++ para Python do código da biblioteca, além de necessitar da utilização de dependências equivalentes. Este processo pode abrir margem para inconsistências na comunicação, não representando também impacto significativo no custo final da aplicação pelo uso da Arduino pro mini.

A comunicação entre as placas feitas por serial, poderia ser implementada de duas formas, pelos pinos de TX e RX de ambas as placas ou por porta USB, ambas as formas entregam os mesmos resultados. A escolha pela comunicação via USB foi

pelo fato do conector estabelecer uma conexão física mais resistente, evitando possíveis falhas em toques acidentais no equipamento.

Figura 6 - Dispositivo responsável e processamento dos dados coletados.



Fonte: O autor, 2023.

A implementação da parte de SW do sistema foi dividida em 3 códigos distintos. Uma camada responsável pela coleta dos dados e executada pelas 2 placas ESP32, feita em C++, responsáveis por coletar e transmitir os dados coletados para o módulo principal da rede. Uma camada responsável por receber, decodificar e enviar para a terceira etapa as mensagens enviadas pelos módulos anteriores.

Uma terceira camada de código referente ao gateway executado pela raspberry pi, feito em python 3, e responsável por separar, converter os dados da coleta e salvar estes de 3 formas:

- Um arquivo de log com toda a comunicação recebida, para possíveis casos de debug;
- O banco de dados PostgreSQL local, uma camada de redundância que disponibiliza os dados em uma eventual perda de conexão com a rede e pode ser utilizado como backup, acessado pelo Metabase executado pela Raspberry Pi;

- O banco de dados PostgreSQL remoto, responsável por ser a fonte principal de dados para o Metabase disponibilizado ao usuário.

3.4. Código Implementado:

Para a implementação, foram utilizados 3 etapas de código, a primeira feita para os dispositivos ESP32 responsáveis pela coletar dos dados referentes ao nível de gás, possuindo o comportamento da seguinte forma:

1. Inicia comunicação Serial com o módulo LoRa;
2. Analisa o valor analógico referente ao output dos dois sensores;
3. Armazena os valores analisados e incrementa o contador de medições;
4. Entra em repouso por 1 minuto;
5. Após completar 5 medições, o pacote é montado e transmitido, e o armazenamento das medições juntamente com o contador são reiniciados.

Para realizar a transmissão do pacote, foi implementada a função `enviaPacote`, que com base no datasheet [19] e na documentação da biblioteca oficial dos módulos LoRaMESH, forma a mensagem, referenciando como destino o módulo mestre da rede, que possui o identificador número 0, informa o endereço de operação de 0x70, que está no endereçamento do módulo LoRaMESH destinado a comandos da própria aplicação, indicando que a mensagem será entregue de forma integral pela comunicação serial do módulo receptor, ficando sob total responsabilidade da aplicação executar o seu comando. As informações enviadas estão contidas no `bufferPayload`, onde é criada uma estrutura em strings com os dados devidamente organizados em tuplas e com separadores, sendo este por sua vez convertido para bytes pela função de `PrepareFrameCommand` da biblioteca oficial, e todo o pacote é enviado pela função de `SendPacket`.

Figura 7 - Código em C++ do dispositivo coletor executado por ambas as ESP32 - LoRa

```

#include "LoRaMESH.h"
#include <SoftwareSerial.h>
#define MQ4Pin 13
#define MQ135Pin 12
#define GasMax 5

SoftwareSerial SerialCommand(16, 17);
LoRaMESH lora(&SerialCommand);
int contadorGas = 0;
int MedidaMQ4[GasMax];
int MedidaMQ135[GasMax];
uint8_t ID = 1;

void enviaPacote();

void setup() {
  Serial.begin(115200);
  delay(2000);
  SerialCommand.begin(9600);
  lora.begin(true);
}

void loop() {
  int valorMQ4 = analogRead(MQ4Pin);
  int valorMQ135 = analogRead(MQ135Pin);
  MedidaMQ4[contadorGas] = valorMQ4;
  MedidaMQ135[contadorGas] = valorMQ135;
  Serial.println(valorMQ4);
  Serial.println(valorMQ135);
  contadorGas++;
  if(contadorGas == 5){
    contadorGas = 0;
    enviaPacote();
  }
  delay(100000);
}

void enviaPacote(){
  uint8_t b = 0;
  uint8_t bufferPayload[MAX_PAYLOAD_SIZE];
  uint8_t idx;
  String msg = "Gas(" + String(lora.localId) + "):";
  for(idx = 0; idx < GasMax; idx++){
    String parGas = "(" + String(MedidaMQ4[idx]) + "," + String(MedidaMQ135[idx]) + ")";
    msg += parGas;
  }
  msg += ";";

  Serial.println(msg);
  for(idx = 0; idx < msg.length(); idx++){
    bufferPayload[idx] = msg[idx];
  }
  lora.PrepareFrameCommand(0, 0x70, bufferPayload, msg.length());
  lora.SendPacket();
}

```

Fonte: O autor, 2023

A segunda etapa da aplicação fica responsável por receber o pacote LoRa enviado, e é executado pela Arduino pro mini, onde nesta etapa a execução se dá:

1. Inicia comunicação Serial com o módulo LoRa;
2. É alocado espaço em memória para realização da leitura dos pacotes;
3. É chamada a função `ReceivePacketCommand` da biblioteca oficial, que é responsável por receber todo o pacote de comunicação e armazenar no espaço alocado;
4. Os valores em bytes são convertidos para um array de char;
5. O array do payload é lido em partes e toda a mensagem é concatenada em uma nova estrutura até a chegada do separador ‘;’ ;
6. A mensagem é enviada via serial para a Raspberry;

Para esta implementação, por se basear em string, o tamanho da mensagem varia, uma vez que os valores de cada parâmetro podem variar de 1 a 4 dígitos, assim precisando só do separador para definir o final da mensagem. Da mesma forma, o delay de 50ms ao final da execução evitou que algumas interferências fossem recebidas ao final da leitura de uma mensagem.

Figura 8 - Código em C++ do gateway executado pela arduino pro mini

```

#include "LoRaMESH.h"
#include <SoftwareSerial.h>

SoftwareSerial SerialCommand(3, 4);
LoRaMESH lora(&SerialCommand);
uint8_t ID = 0;

void setup() {
  Serial.begin(115200);
  delay(2000);
  SerialCommand.begin(9600);
  lora.begin();
}

void loop() {
  uint8_t b = 0;
  uint8_t bufferPayload[MAX_PAYLOAD_SIZE];
  uint8_t* payloadSize;
  uint16_t* id;
  uint8_t* command;
  if (lora.ReceivePacketCommand(id, command, bufferPayload, payloadSize, 1)){
    char* buff = (char *) bufferPayload;
    String buffStr = String(buff);
    String msg = "";
    for (int i = 0; buffStr[i] != ';'; i++){
      msg += buffStr[i];
    }
    Serial.println("Recebido: " + msg);
  };
  delay(50);
}

```

Fonte: O autor, 2023

A terceira camada de software da aplicação é o código em python executado pela Raspberry pi, possuindo o papel de receber as mensagens após a decodificação da camada anterior e a comunicação Serial com a Arduino. Esta camada também armazena toda a comunicação recebida em um arquivo de log, e após isso, seleciona apenas a mensagem referente ao pacote com informações do nível dos gases, onde para cada mensagem de interesse:

1. A aplicação calcula o horário de recebimento;
2. Separa os valores da mensagem e converte os dados para seus respectivos valores numéricos e armazena em uma nova estrutura temporária;

3. Calcula os valores dos gases em PPM (parte por milhão), com base nos valores de referência do datasheet;
4. Calcula o horário aproximado de coleta, com base no horário de recebimento da mensagem e na ordem em que cada par foi colocado na mensagem;
5. Estabelece uma conexão com as duas instâncias do banco de dados, a local e a remota;
6. Executa a operação de inserir os valores no banco de dados para cada uma das instâncias.

Figura 9 - Código em python do gateway executado pela raspberry pi

```

import RPi.GPIO as GPIO
import time
import serial
from datetime import *
import psycopg2

#Configura a serial e a velocidade de transmissao
ser = serial.Serial("/dev/ttyAMA0", 115200)

GPIO.setmode(GPIO.BOARD)
fazer = True

while(fazer):
    dataStr=""
    data = ser.readline()
    dataStr = str(data)
    dataStr = dataStr[2:len(dataStr)-5]
    idData = 0
    log = open("log.txt", "a")
    log.write(dataStr)
    log.close()
    try:
        if("Recebido" in dataStr):
            now = datetime.now()
            delta = timedelta(minutes=1)

            gasData = dataStr.split()[1]
            gasData = gasData.split(':')

            idStr = gasData[0]
            gasId = int(idStr[4:len(idStr)-1])

            pairsStr = gasData[1]
            pairs = pairsStr[1:len(pairsStr)-1].split(',')

            pairsVal = []
            for p in pairs:
                pair = p.split(',')
                p0 = int(pair[0])
                p1 = int(pair[1])
                pairsVal += [(p0,p1)]
            print(pairsVal)
            print(gasId)
            print(pairs)
            print(dataStr)

            con = psycopg2.connect(host='localhost', database='tcc',user='xxxxxxx', password='xxxxxxx')
            conRemoto = psycopg2.connect(host='remote_host', database='tcc',user='xxxxxxx', password='xxxxxxx')
            for p in reversed(pairsVal):
                cur = con.cursor()
                curRemoto = conRemoto.cursor()

                metanoBruto = p[0]
                amoniaBruto = p[1]

                metanoPPM = str(int(metanoBruto*(10000/1024)))
                amoniaPPM = str(int(amoniaBruto*(10000/1024)))

                sql = "insert into monitoramento(dia,hora,dispositivo,metanoBruto,metanoPPM,amoniaBruto,amoniaPPM)
values ('"
                sql += str(now.date())+"',"+str(now.time())
[0:8]+"',"+str(gasId)+"',"+str(metanoBruto)+"',"+str(metanoPPM)+"',"+str(amoniaBruto)+"',"+str(amoniaPPM)+"")
                cur.execute(sql)
                curRemoto.execute(sql)
                con.commit()
                curRemoto.commit()
                now = now - delta

            except:
                print("Decode Error")

```

Fonte: O autor, 2023

Após a execução de todas estas camadas os dados se encontram devidamente inseridos em ambos os bancos de dados, onde por estarem acoplados as duas

instâncias do Metabase, também mantém o painel de monitoramento atualizado, em tempo real e cabendo apenas ao usuário a recarga da página de exibição do painel, ou a manutenção da configuração de atualização automática para que as informações possam ser acompanhadas.

O painel do Metabase foi construído com a exibição dos dados ao longo do tempo, as contagens de coletas e a média e desvio padrão destes, agregados em unidades de tempo, de dias, horas e minutos. A formação deste painel permite que usuários não autenticados na aplicação do metabase acompanhe a evolução dos dados disponibilizados em tempo real, e para usuários autenticados e com o nível de permissão para edição, a possibilidade de realizar consultas de novas informações e edição do painel para que estas possam ser acrescentadas.

3.5 Ambiente De Testes:

Para a execução destes experimentos, o ambiente online foi utilizado um servidor hospedado na plataforma Heroku, com o plano Basic, possuindo 512 MB de memória Ram e 1 CPU compartilhado, sendo este o ambiente responsável por todo o processamento e hospedagem do Metabase. Como o banco de dados remoto, foi utilizado um plano basic do Heroku Postgres, possuindo até 10,000,000 linhas de limite e 10 GB de armazenamento máximo.

O ambiente no qual os dispositivos foram instalados se trata de um barracão de confinamento no sistema de Compost Barn, situado na Fazenda Baronesa, localizada na cidade de Altinho/PE, possuindo dimensões de 78 por 48 metros, com um corredor de alimentação de 5 metros, possuindo duas pistas de trato, uma em cada lado do barracão, com 4,5m de largura por 72m de comprimento e duas camas, uma de cada lado, de 14m de largura por 72m de comprimento, constituídas por serragem de madeiras diversas e sendo responsável pela compostagem dos dejetos produzidos, também responsável por produzir gases a serem analisados.

Os dispositivos responsáveis pela coleta foram instalados em dois pilares do barracão analisado, presos por dois fios de nylon em cada uma das protoboards e estes fios por sua vez presos aos respectivos pilares onde foram instalados por meio

de um gancho feito de arame galvanizado. Desta forma, os dispositivos se mantiveram fixos e puderam ser retirados para eventuais manutenções.

O ambiente pode ser visto nas Figuras 11 e 12, e um dispositivo instalado pode ser encontrado na Figura 10.

Figura 10 - Dispositivo responsável por mensurar instalado no barracão do confinamento



Fonte: O autor, 2023

Figura 11 - Barracão de confinamento - visão frontal



Fonte: O autor, 2023

Figura 12 - Barracão de confinamento - visão lateral



Fonte: O autor, 2023

4.RESULTADOS

Ao todo, foram enviados 115 pacotes de medições, totalizando 575 análises feitas e inseridas no banco de dados. Algo perceptível são as discontinuidades em alguns momentos das coletas, isto ocorreu devido a descarga das baterias, onde no cenário de maior duração, com os módulos em modo de operação tipo A e janela de 5 segundos foi possível transmitir 55 pacotes, em comparação aos 40 pacotes ao se utilizar os módulos em modo de operação tipo C.

Com a utilização do Metabase, é possível extrair informações relevantes sobre o comportamento do ambiente monitorado, como a evolução ao longo do tempo (Figura 13), evolução das médias das medições (Figura 14) e a quantificação e dispersão dos dados (Figura 15). Desta forma, é possível notar variações conforme diminuições de animais no local, por volta das 9 horas da manhã do dia 3 de setembro. É visto uma diminuição em decorrência do acionamento do sistema de ventilação, por volta das 19 horas da noite do dia 2 de setembro.

Outros dados perceptíveis são alguns valores fora de escala, onde um conjunto de 3 pacotes foi enviado com valores analógicos acima de 1024, fora da escala esperada. Estes 3 pacotes foram os últimos enviados pelo dispositivo antes de sua descarga completa, podendo indicar falhas no gerenciamento de energia em determinados níveis de alimentação.

É possível perceber tendências nos gases, onde observando variações nos resultados em momentos de acionamento do sistema de ventilação ao final do primeiro conjunto de medições, resultado de uma dissipação do acúmulo dos gases. Existe a tendência de uma queda nos dados ao final do último conjunto de medições, em um momento no qual os animais estavam sendo retirados do confinamento para a ordenha, resultando em uma menor produção dos mesmos gases.

Figura 13 - Painel do Metabase com os dados coletados - dados distribuídos por tempo



Fonte: O autor, 2023

Figura 14 - Painel do Metabase com os dados coletados - Dados agregados



Fonte: O autor, 2023

Figura 15 - Painel do Metabase com os dados coletados - Estatísticas gerais



Fonte: O autor, 2023

5. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho entrega um sistema de baixo custo capaz de indicar tendências em relação aos níveis dos gases analisados, sendo possível que com um baixo investimento, novos produtores insiram tecnologias no seu processo de tomada de decisão. O sistema foi capaz de detectar variações de níveis dentro de um mesmo dia, mostrando impactos de ações na rotina exercida no manejo do gado.

Isto abre a possibilidade do produtor modificar o manejo e analisar o impacto que tais mudanças tem em seu ambiente, com uma coleta em tempo real, disponibilizada online, para que outros profissionais possam auxiliar na análise, e com a disponibilidade na rede local, em caso de indisponibilidade de conexão.

Como pontos de melhoria para o futuro, é possível perceber o impacto do consumo rápido da bateria, exigindo recargas destas com uma alta frequência, necessitando de aprimoramento com o intuito de viabilizar tal sistema. É de interesse análises de como encontrar uma frequência menor capaz de realizar a mesma análise com a qualidade desejada, e reduzindo a quantidade de medições desnecessárias, bem como formas de reduzir o consumo do sistema quando não utilizado, como o corte ou redução da alimentação dos sensores de gás.

Outro ponto de melhoria com relação ao consumo energético de tal sistema é a melhoria na comunicação, trabalhando com formatos de dados de menor nível de abstração, e que por sua vez constituem mensagens menores e que consomem menos para a realização da transmissão entre os módulos, assim como uma análise da forma de configuração dos módulos LoRa que devem ser mais eficientes.

Por fim é importante ressaltar que o meio ao qual o equipamento é exposto se apresenta bastante agressivo com tais equipamentos, sendo importante para implementações futuras a utilização de uma proteção mais resistente que os de prototipação, garantindo uma maior sobrevida.

REFERÊNCIAS

[1] Produção de Leite IBGE:

<https://www.ibge.gov.br/explica/producao-agropecuaria/leite/br> Acesso em: 01 ago. 2023

[2] Projeções para os próximos anos:

<https://www.gov.br/agricultura/pt-br/assuntos/politica-agricola/todas-publicacoes-de-politica-agricola/projecoes-do-agronegocio/projecoes-do-agronegocio-2022-2023-a-2032-2033.pdf/>

Acesso em: 01 ago. 2023

[3] Plano de redução de emissão de carbono:

<https://www.gov.br/agricultura/pt-br/assuntos/noticias/com-tecnologias-de-producao-sustentavel-plano-abc-pretende-reduzir-emissao-de-carbono-em-mais-de-1-bilhao-de-toneladas>

Acesso em: 01 ago. 2023

[4] Plano ABC+:

<https://www.gov.br/agricultura/pt-br/assuntos/sustentabilidade/plano-abc/arquivo-publicacoes-plano-abc/abc-portugues.pdf> Acesso em: 01 ago. 2023

[5] Crescimento da produção de leite em 10 anos:

<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/26457-produtividade-de-leite-cresce-62-em-11-anos-producao-chega-a-30-bilhoes-de-litros>

Acesso em: 03 ago. 2023

[6] Censo Agropecuário IBGE:

<https://www.ibge.gov.br/estatisticas/economicas/agricultura-e-pecuaria/21814-2017-censo-agropecuario.html> Acesso em: 03 ago. 2023

[7] Confinamento de gado leiteiro ganha força:

<https://agroemdia.com.br/2022/07/13/confinamento-total-de-vacas-em-lactacao-ganha-forca-no-brasil-diz-cepea/> Acesso em: 03 ago. 2023

[8] Transição para o confinamento demanda planejamento financeiro e metas de produtividade:

<https://cepea.esalq.usp.br/upload/revista/pdf/0169143001657637729.pdf> Acesso em: 03 ago. 2023

[9] Corrosão da amônia em estruturas de concreto armado:

<https://core.ac.uk/download/pdf/234677736.pdf> Acesso em: 09 ago. 2023

[10] Corrosão de amônia em estruturas de concreto armado e metálico em fazendas:

<https://www.mdpi.com/2076-3417/10/3/990> Acesso em: 10 ago. 2023

[11] IOT Based Industrial Plant Safety Gas Leakage Detection System:

<https://ieeexplore.ieee.org/abstract/document/8777463/> Acesso em: 12 ago. 2023

[12] Wireless Gas Leak Detection and Localization:

<https://ieeexplore.ieee.org/abstract/document/7029005> Acesso em: 12 ago. 2023

[13] Context-Aware Monitoring and Control of Ventilation Rate in Indoor Environments Using Internet of Things:

<https://ieeexplore.ieee.org/abstract/document/9351545/> Acesso em: 15 ago. 2023

[14] In Situ Measurement of Methane Flux from Residential Compost Using Automated Static Chamber and Electrochemical Sensor:

<https://ieeexplore.ieee.org/abstract/document/8666383> Acesso em: 15 ago. 2023

[15] IoT System based Forecasting and Modeling Exceedance Probability and Return Period of Air Quality using Extreme Value Distribution:

<https://ieeexplore.ieee.org/abstract/document/8706035> Acesso em: 18 ago. 2023

[16] Biblioteca LoRaMESH Radioenge: <https://github.com/Radioenge/LoRaMESH> Acesso em: 28 ago. 2023

[17] LoRa Technology - An Overview: <https://ieeexplore.ieee.org/abstract/document/8474715> Acesso em: 17 set. 2023

[18] Synchronous LoRa Mesh Network to Monitor Processes in Underground Infrastructure:

<https://ieeexplore.ieee.org/abstract/document/8703036> Acesso em 17 set. 2023

[19] Datasheet módulo LoRaMESH Radioenge:

https://www.radioenge.com.br/wp-content/uploads/2021/08/Manual_LoRaMESH_jul_2023.pdf Acesso em 29 set. 2023