

Livestock Monitoring Prototype Implementation and Validation

Implementação e Validação de um Protótipo para Monitorização de Gado

Vitor M. T. Aleluia¹, Vasco N. G. J. Soares^{1,2*}, João M. L. P. Caldeira^{1,2}, Pedro D. Gaspar^{3,4}

Resumo: This paper presents the proposal, implementation and validation of a low cost fault-tolerant functional prototype for livestock monitoring. This prototype uses IoT devices, ESP8266 and ESP32, creating a mesh network, managed by the painlessMesh library, with WiFi and LoRa technologies. It allows, for instance, the collection of vital signs from animals. In comparison with the traditional method of livestock examination, this cost-efficient approach reduces manual labor and saves working time. It also improves animal health, increases profits and decreases the environmental footprint.

Keywords: Prototype — Wireless mesh networks — Livestock monitoring — PainlessMesh

Resumo: O presente artigo apresenta a proposta, implementação e validação de um protótipo funcional tolerante a falhas e de baixo custo para a monitorização do gado. Este protótipo recorre à utilização de dispositivos IoT, ESP8266 e ESP32, criando uma rede mesh, gerida pela biblioteca painlessMesh, com tecnologias de comunicação WiFi e LoRa. Este protótipo, possibilita, por exemplo, a recolha de sinais vitais dos animais. Em comparação com o método tradicional de inspeção de gado, esta abordagem eficiente e de baixo custo, reduz o trabalho manual e poupa tempo de trabalho. Contribuindo ainda para melhorar a saúde animal, aumentar os lucros e diminuir a pegada ambiental.

Palavras-Chave: Protótipo — Redes em malha sem fios — Monitorização de gado — PainlessMesh

¹ Instituto Politécnico de Castelo Branco (IPCB), Castelo Branco - Beira Baixa, Portugal

² Instituto de Telecomunicações, Covilhã - Beira Baixa, Portugal

³ Departamento de Engenharia Eletromecânica, Universidade da Beira Interior (UBI), Covilhã - Beira Baixa, Portugal

⁴ Centro de Ciência e Tecnologias Mecânicas e Aeroespaciais (C-MAST), Covilhã - Beira Baixa, Portugal

*Corresponding author: vascog.soares@ipcb.pt

DOI: <http://dx.doi.org/10.22456/2175-2745.127207> • Received: 16/09/2022 • Accepted: 14/01/2023

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

1. Introdução

Os sistemas e dispositivos estão a tornar-se cada vez mais conectados devido ao aumento da popularidade da Internet das coisas (em Inglês *Internet of Things* (IoT)). As automatizações utilizando dispositivos IoT estendem-se a várias áreas tais como, mas não limitadas a, cidades inteligentes, transportes, saúde e indústria. A Internet das Coisas é uma das tecnologias que mais pode viabilizar a agricultura de precisão, por exemplo na gestão e produção de gado.

A gestão do gado é essencial para que seja assegurada a qualidade de vida do animal, enquanto são preservados os interesses financeiros dos agricultores. Para determinados animais a qualidade do produto (carne ou leite) depende da habilidade e experiência do agricultor em monitorizar e controlar o processo. A tendência crescente na produção animal, tornou este processo mais exigente e trabalhoso, uma vez que, no método tradicional de monitorização de gado é necessário

observar cada animal fisicamente para verificar o seu estado de saúde e bem-estar [1].

A utilização de sensores, juntamente com dispositivos IoT pretende dar uma resposta ao problema da monitorização de gado em larga escala. No passado, foram propostas várias soluções para a monitorização de gado, como é o caso de Abdullahi et al. [2] e Germani et al. [3] onde ambas fazem uso de plataformas Arduino [4] e RaspberryPi [5], e da tecnologia de comunicação LoRaWAN [6]. As redes LoRa são redes de comunicação de longo alcance, baixo consumo energético e com baixa velocidade de transferência de dados. Existem outras redes de longo alcance como, por exemplo, SigFox [7] e NB-IoT [8], no entanto, as redes LoRaWAN possibilitam a criação de redes privadas e integração com outros serviços como, por exemplo, The Things Network [9].

O presente trabalho parte das conclusões apresentadas num artigo anterior dos mesmos autores [10], que analisou as soluções existentes para os diferentes tipos de identifica-

ção e monitorização animal. Nesse artigo foram identificadas as abordagens, os desafios e as oportunidades no uso de dispositivos IoT para monitorizar o gado. Este artigo pretende apresentar a proposta de uma infraestrutura de comunicações de dados tolerante a falhas para uso na monitorização de gado. Para tal foi proposta uma arquitetura de rede tolerante a falhas, através da utilização de módulos de comunicação LoRa e WiFi. O mecanismo de redundância proposto é baseado no princípio de funcionamento do protocolo *Virtual Router Redundancy Protocol (VRRP)* [11]. Esta solução visa demonstrar a viabilidade do uso desta tecnologia no suporte à recolha contínua (sem interrupções) de parâmetros biológicos dos animais, usando diversos sensores. O uso desses sensores e a sua parametrização está fora do âmbito deste trabalho e será dependente do caso prático da utilização em concreto desta solução.

O presente artigo está estruturado em quatro secções. A Secção 2 apresenta os projetos de monitorização de gado relacionados com a monitorização de sinais vitais e com a localização do animal. Na Secção 3 é apresentada a proposta e implementação de um protótipo para monitorização de gado tolerante a falhas. Na Secção 4 é validado o protótipo, discutindo-se os resultados observados. Por fim a Secção 5 conclui o artigo e apresenta o trabalho futuro.

2. Monitorização de Gado

Esta secção apresenta trabalhos existentes na literatura que visam a temática da monitorização dos sinais vitais, permitindo detetar, tratar ou prevenir doenças e lesões, salvaguardando a saúde dos animais.

O projeto apresentado em [12] foi concebido com o objetivo de monitorizar animais de quinta como, por exemplo, vacas leiteiras, cabras, ovelhas, porcos e cavalos, de modo a detetar anomalias com recurso a sensores. Através da monitorização de gado é possível detetar alterações físicas prevendo a fertilidade, ciclo éstrico, stress térmico e a febre do leite. Assim, pode-se prever doenças que levem a consequências graves na vida do animal e consequentes perdas económicas. O projeto é composto por uma plataforma de prototipagem BITalino R-IoT [13] que possui WiFi [14], acelerómetros, giroscópios e magnetómetros para registar os movimentos do animal. Posteriormente foi acrescentado um sensor de eletrocardiografia para produzir um eletrocardiograma dos batimentos cardíacos do animal. O protótipo apresentado na Figura 1 funciona através da análise dos dados recolhidos pelos sensores. Caso detete que estes não se encontram dentro de parâmetros normais, é despoletado um alerta enviado por correio eletrónico ao agricultor. Os resultados apresentados no estudo revelam anormalidades na frequência cardíaca dos bovinos, sendo esta justificada pelo stress do animal durante os testes.

A solução produzida pela empresa Cowlar [15] para monitorizar vacas, permite detetar doenças, otimizar a alimentação, gerir do stress e auxilia no aumento da taxa de reprodução. O dispositivo mede a temperatura do animal, os níveis de



Figura 1. Bovino com o protótipo desenvolvido por Reigones et al. Fonte: Adaptado de [12].

atividade e comportamento, que detetam se a vaca esteve a comer ou a dormir. A solução é composta por uma caixa que é posicionada na parte anterior da cabeça da vaca (Figura 2) e um *router* alimentado através de um painel solar, responsável por manter a conexão constante com os colares. Os dados recolhidos pelos colares são enviados para o *router* que, por sua vez, através de uma conexão à rede móvel comunica com a *cloud* da empresa Cowlar. Isto permite o acesso à informação na web através de um painel que mostra todos os dados recolhidos, os respetivos gráficos e conselhos de ações a tomar para aumentar a produtividade e o bem-estar das vacas. Dispõe ainda da funcionalidade de alerta que envia mensagens quando a vaca exibe sintomas de febre.



Figura 2. Solução da empresa Cowlar. Fonte: Adaptado de [15].

A empresa EnviarIoT [16] produziu um sistema para monitorização de animais de quinta, ilustrado na Figura 3, que consiste em sensores para recolha de dados e uma estação para receber os dados. Os sensores utilizados pelo sistema têm como objetivo assegurar o bem-estar dos animais recolhendo indicadores, como por exemplo temperatura, humidade e níveis de amoníaco. Os módulos são eficientes energeticamente, e podem ser alimentados por painéis solares. Após os dados serem recolhidos, estes são enviados para a estação colocada

na quinta. Esta por sua vez, envia os dados através da rede móvel para a *cloud* da empresa (EnviraIoT) ou de terceiros através do 3G, 4G e LoRa. As redes LoRaWAN, 3G e 4G têm a vantagem de ser redes de longo alcance pelo que apenas é realizado um salto entre os dispositivos responsáveis por medir os dados e o *gateway* (estação central).



Figura 3. Sistema de monitorização EnviraIoT. Fonte: Adaptado de [16].

O NFCGROUP desenvolveu e comercializa um bolo ruminal [17] apresentado na Figura 4 para monitorizar a temperatura de bovinos em larga escala. A aplicação do bolo ruminal é feita através de um aplicador próprio que é colocado no início do trato digestivo do animal e, por sua vez, por ação gravítica acaba por se depositar no rúmen, local onde permanecerá até ao fim da vida do animal. Embora o bolo ruminal fique depositado para sempre, a sua autonomia em bateria é de 5 anos. Os dados recolhidos da temperatura pelo bolo ruminal são enviados para um *gateway* instalado localmente. Este é responsável por enviar os dados para a *cloud*, aos quais o utilizador poderá aceder através de um computador ou *smartphone*.



Figura 4. Bolo ruminal Activeherd. Fonte: Adaptado de [17].

Para além das soluções para monitorização dos sinais vitais de gado, têm vindo a ser propostas soluções para monitorização da sua localização. Esta necessidade deve-se ao facto de os animais ruminantes necessitarem de grandes áreas para pastar.

No projeto desenvolvido em [18] foi apresentada uma solução de baixo custo para a monitorização da localização de gado. O sistema faz uso das tecnologias GPS [19], Bluetooth Low Energy (BLE) [20] e uma rede de longo alcance e baixo consumo energético designada Sigfox [7]. A localização exata é obtida através da tecnologia GPS, enquanto o BLE permite a localização relativa a outros dispositivos em ambientes fecha-

dos. Esta solução recorre a dois dispositivos diferentes. Um é responsável por obter a localização exata através do GPS e o outro obtém a localização relativa ao primeiro através do BLE. O primeiro dispositivo é um colar composto por um módulo GPS, um módulo Sigfox e um módulo BLE (Figura 5). O segundo é um brinco (de gado) composto por um módulo BLE. Uma vantagem desta solução é a eficiência energética. Segundo o autor, o colar GPS tem uma autonomia de 365 dias e o brinco BLE dura 280 dias.

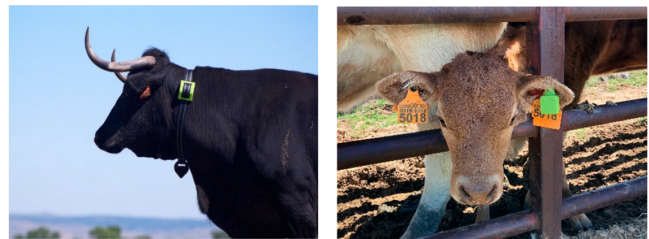


Figura 5. Solução desenvolvida por Maroto-Molina e outros autores. Fonte: Adaptado de [18].

A empresa mOOvement [21] possui uma solução para monitorização da localização de gado. O dispositivo é apresentado na forma de brinco como se verifica na Figura 6, que possui as tecnologias GPS e LoRa, para deteção da localização e envio da informação a longa distância. O brinco opera através de baterias, no entanto, dado o seu consumo energético reduzido, estas podem ser carregadas através de um painel fotovoltaico presente no colar. Os dados recolhidos são enviados para a plataforma da empresa através da rede LoRa que permite ao utilizador o acesso aos mesmos e a receção de alertas.



Figura 6. Dispositivo de monitorização de localização mOOvement. Fonte: Adaptado de [22].

O trabalho apresentado em [23] consiste num sistema para recolha de localização do gado em tempo real através de redes sensores sem fios. As vacas tendem a pastar em áreas alargadas e aquando do fim do pasto ficam em repouso. No entanto, se o repouso for demasiado prolongado, isto serve como indicador do estado de saúde da vaca. Através da monitorização da localização é possível detetar as vacas que necessitem de ser examinadas fisicamente. A solução desenvolvida (Figura

7) é um colar composto por um microcontrolador, um módulo GPS, um módulo 3G e um módulo de comunicação sem fios de curta distância. O módulo GPS é responsável por obter a localização exata do animal, enquanto os módulos de redes sem fios de curta distância permitem a partilha dos dados para outros colares próximos. O envio da informação para a plataforma de *cloud*, é feito através do módulo 3G. Em cada manada é escolhido um líder, este é o responsável por agregar as informações dos outros colares e enviar através da rede 3G. Caso os animais se dividam e percam acesso ao líder então será declarado um líder temporário, já que cada colar tem um módulo 3G e possui acesso à plataforma *cloud*.



Figura 7. Vaca com um colar desenvolvido por Park e colegas. Fonte: Adaptado de [23].

O projeto ZebraNet [24] foi desenvolvido com o intuito de monitorizar os movimentos de animais selvagens, para estudar as interferências de fatores exteriores como a temperatura e a presença de humanos. A monitorização de animais selvagens é complexa atendendo ao facto de estes não estarem habituados a permanecer confinados a um local. Assim, foram utilizados módulos de comunicação de longa distância (e.g., rádio). Adicionalmente, os dados necessitam de ser recolhidos sem intervenção humana. O colar desenvolvido para este projeto e responsável por recolher os dados é composto por um módulo GPS, um controlador com CPU e RAM, e duas antenas de rádio, uma de curto alcance e outra de longo alcance. Para além disso, o colar possui também uma bateria, um painel solar e um circuito para gestão energética. O módulo GPS permite obter a localização do animal, enquanto a antena de curto alcance comunica com os outros colares, por outro lado, a antena de longo alcance comunica com uma estação de rádio para transmitir os dados recolhidos. Para que os colares sejam energeticamente eficientes, foi adotada uma estratégia que permite a hibernação do colar durante um determinado período, o que se traduz numa poupança significativa de energia. O colar desenvolvido no âmbito do projeto ZebraNet é apresentado na Figura 8.



Figura 8. Zebra com o colar desenvolvida no projeto ZebraNet. Fonte: Adaptado de [25].

3. Proposta e Implementação do Protótipo

Esta secção descreve a proposta e a implementação do protótipo funcional de baixo custo com tolerância a falhas para monitorização de gado, cuja arquitetura é apresentada na Figura 9. A secção encontra-se dividida em duas partes, na primeira é apresentado o hardware utilizado no desenvolvimento do protótipo e na segunda é descrita o software implementado.

A arquitetura é composta por colares com sensores, dispositivos IoT e módulos de comunicação LoRa e WiFi.

A especificação LoRaWAN [6] é um protocolo para redes sem fios de longo alcance (cinco quilómetros em áreas urbanas e quinze quilómetros em áreas rurais [26]) e baixo consumo energético, concebida para dispositivos que operam com baterias, como é o caso de muitos dispositivos IoT. A sua desvantagem, em relação a outras redes sem fios, reside na velocidade de transferência de dados que varia entre 0,3 kb/s a 50 kb/s. As redes LoRa [26] seguem uma topologia em estrela, onde todos os dispositivos estão conectados a um *gateway*, para o qual são enviados os dados.

As redes WiFi [27] são redes locais sem fios (em Inglês *Wireless Local Area Networks*) como o nome indica, são redes que utilizam frequências de rádio não licenciadas. Ao contrário das redes LoRa, as redes WiFi foram desenhadas para dispositivos mais robustos como é o caso dos *smartphones* e computadores, que necessitam de maior largura de banda do que os dispositivos IoT. As redes WiFi possuem um alcance mais baixo e um consumo energético superior às redes LoRa, no entanto proporcionam velocidades maiores de transferência de dados. Além disso, as redes WiFi podem adotar uma topologia em estrela à semelhança das redes LoRa, ou em malha onde todos os dispositivos estão interligados entre si e reencaminham as mensagens entre eles em vez de um dispositivo central (*gateway/router*).

Inicialmente, foi pensada numa arquitetura apenas com módulos de comunicação LoRa. No entanto, considerando o facto de que um rebanho pode ser constituído por milhares de animais, se todos possuírem colares a monitorizar e a enviar dados ao mesmo tempo, prevê-se que a rede ficaria

congestionada devido ao elevado volume de dados. O estudo apresentado em [28] demonstrou que a transmissão de elevado número de dados é apenas possível num ambiente controlado e com um número reduzido de dispositivos.

Face as limitações da rede LoRa, em complemento considerou-se a utilização de uma rede WiFi com uma topologia em malha que, por um lado torna a rede mais robusta quanto mais dispositivos forem ligados a ela, e por outro em caso de falha num dos dispositivos os restantes reorganizam-se entre si e reencaminham a mensagem até ao destino [29]. Assim, aumenta o alcance da rede, a probabilidade de entrega de mensagens, a confiabilidade e a tolerância à falhas.

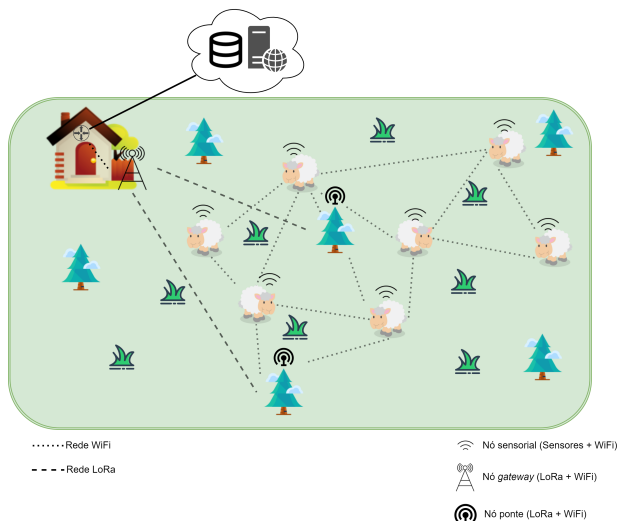


Figura 9. Diagrama da arquitetura do protótipo.

A arquitetura ilustrada na Figura 9 está dividida em três partes. Cada uma possui dispositivos denominados por nós, com diferentes funções e características. Os nós sensoriais referem-se aos animais com os colares, que possuem um dispositivo IoT que agrega um conjunto de sensores responsáveis pela monitorização, que podem ser escolhidos em função do cenário ou da espécie do animal (e.g., temperatura do animal, temperatura e humidade do ar). Os nós ponte, que podem ser um nó raiz ou um nó backup, fazem a ligação entre a rede WiFi e a rede LoRa, são responsáveis por recolher todos os dados transmitidos pelos nós sensoriais e enviá-los para o nó gateway que, por sua vez, os enviará para uma plataforma *cloud*. Através de uma plataforma *cloud* serão disponibilizados os dados recolhidos pelos nós sensoriais ao agricultor, que poderá aceder a estes em vários dispositivos (e.g., computador ou *smartphone*). O desenvolvimento desta arquitetura tem vários constrangimentos, nomeadamente o consumo energético dos nós sensoriais. Visto serem nós móveis, necessitam de ter uma bateria e como a monitorização é realizada ciclicamente é imperativo economizar o máximo de energia. A Tabela 1 descreve os vários desafios e respetivas hipóteses consideradas no desenvolvimento do protótipo. Ao longo das próximas secções serão identificadas e descritas as soluções que dão resposta a cada um dos desafios identificados considerando as

várias hipóteses em análise.

3.1 Componente de Hardware

Procurando minimizar custos, o protótipo apresentado tem por base dispositivos com microcontroladores ESP8266 e ESP32, com suporte WiFi, compatíveis com a biblioteca *mesh* [30] e que permitem a utilização de serviços HTTP (*Hypertext Transfer Protocol*) para comunicação com APIs (*Application Programming Interface*) [31]. Cada dispositivo tem um custo de aquisição de aproximadamente cinco euros.

Para os nós sensoriais foi escolhido o dispositivo D1 Mini Pro [32] com microcontrolador ESP8266, que possui um módulo WiFi embutido, 16 MB de armazenamento flash e interface para ligação com uma bateria externa. É um dispositivo modular (onze pins digitais e um analógico), compacto (48x25.4mm) e leve (4.5g).

Os nós ponte, que asseguram a ligação entre as duas redes WiFi e LoRa, necessitam de dois módulos de comunicação. Para assegurar a ligação entre os dispositivos LoRa é necessário um *transceiver* LoRa que utiliza frequências de rádio sem licença e de acesso livre. Portugal e na maioria dos países da Europa, têm as frequências 433Mhz e 863Mhz até 870Mhz livres para utilização [33]. No entanto, estas frequências têm diferentes características [34], a frequência 433Mhz tem maior alcance, enquanto as frequências entre 863Mhz e 870Mhz têm maior taxa de sucesso na entrega e recessão de pacotes. Para os nós ponte foi escolhido o dispositivo TTGO [35] que integra um microcontrolador ESP32 de 160Mhz (duas vezes mais rápido que o ESP8266 [36]), possui 16MB de memória flash, módulo de WiFi embutido, módulo e antena LoRaWAN e 32 pins digitais. É, à semelhança do dispositivo D1 Mini Pro, leve e compacto, embora ligeiramente maior. Dado que estes dispositivos utilizam duas redes *wireless*, é previsível que o seu consumo seja maior em comparação àqueles que apenas usam uma, então optou-se por colocar os nós ponte numa posição fixa (e.g., árvore ou estábulo), simplificando a arquitetura (ver Tabela 1, D1).

Para o nó gateway, que serve de intermediário entre a rede LoRa e a plataforma *cloud*, foi escolhido o dispositivo D1 Mini Pro e o *transceiver* RFM95W [37], por ser uma solução de baixo custo e que pode ser ligado ao D1 Mini Pro, no seu lugar, também pode ser utilizado o dispositivo TTGO que possui características semelhantes. A Figura 10 apresenta os dispositivos utilizados para a construção do protótipo.

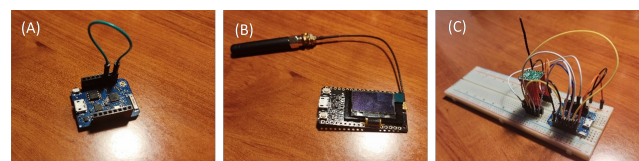


Figura 10. Dispositivos do protótipo: Nó sensorial (A); Nó ponte (B); Nó gateway (C).

Tabela 1. Desafios que podem levar a constrangimentos no protótipo.

Desafios	Hipóteses
Como é delegado o nó raiz? (D1)	<ul style="list-style-type: none"> • O nó raiz é um dos animais? • Um dos nós sensoriais é o nó raiz? • O nó raiz poderá ser um elemento móvel (um trator)? • O nó raiz pode ser aplicado num elemento fixo existente no pasto (uma árvore)?
Como proceder caso existam dois nós raiz? (D2)	<ul style="list-style-type: none"> • Ambos servem de nó raiz? • Apenas um é o nó raiz e o outro nó fica como nó de reserva?
Se o nó raiz falhar, como procedem os outros nós? (D3)	<ul style="list-style-type: none"> • Armazenam os dados? • Verificam a disponibilidade do nó raiz até conseguirem enviar os dados? • Ignoram os dados que não conseguiram transmitir?
Alguns nós sensoriais podem não conseguir comunicar com o nó raiz ou a rede em malha. (D4)	<ul style="list-style-type: none"> • Tentam reenviar a informação e param quando conseguirem? • Descartam os dados? • Armazenam-nos na memória? • Enviam apenas a leitura mais recente? • Reencaminham a informação para um nó vizinho? • Desligam? E quando voltam a ligar?
Os nós sensoriais conseguem ligar-se à rede e enviam a mensagem, mas não recebem confirmação da receção (ACK). (D5)	<ul style="list-style-type: none"> • Reenviam a mensagem? • Descartam a mensagem?
Se um nó sensorial estiver há muito tempo em atividade. (D6)	<ul style="list-style-type: none"> • Deve adormecer por si próprio? • Aguardar mensagem do nó raiz para adormecer? • Continuar ativo?
Verificar o tempo dos dados, quais as leituras mais recentes e mais velhas? (D7)	<ul style="list-style-type: none"> • Armazenar a informação? • Utilização de marcas temporais nos dados? • Enviar apenas a leitura mais recente?

3.2 Componente de Software

O protótipo utiliza a biblioteca `painlessMesh` [30] desenvolvida para Arduino. Esta biblioteca é compatível com as plataformas que integrem tanto microcontroladores ESP8266 como ESP32. Para além desta biblioteca, são utilizadas bibliotecas complementares ao funcionamento dos módulos de comunicação, sendo a biblioteca WiFi incluída na biblioteca `painlessMesh` e a biblioteca LoRa [38] para os dispositivos que operam com a mesma (nó ponte e nó *gateway*). O nó *gateway* para além da biblioteca WiFi e LoRa, possui também a biblioteca `HTTPClient` [39] para que possa comunicar com a API para onde serão enviados os dados. A programação dos dispositivos foi realizada com recurso à ferramenta PlatformIO [40], uma extensão para o editor de texto Microsoft Visual Studio Code que permite a gestão de bibliotecas, desenvolvimento multiplataforma e sugestões de código.

Procurando dar resposta aos desafios acima apresentados foram desenvolvidos um conjunto de algoritmos. Os comportamentos dos diferentes tipos de nós da rede são descritos nos tópicos seguintes.

Nó Sensorial

O nó sensorial tem como função recolher os dados a partir dos sensores nele instalados (e.g., temperatura corporal, temperatura e humidade do ar), e o envio posterior dos mesmos para um nó ponte. Em caso de falha de comunicação, não armazena as leituras para que os dados enviados sejam consistentes e apenas os mais recentes (ver Tabela 1, D3 e D7). Para evitar que este nó fique sempre ativo e para que seja economizada energia, foi implementada uma estratégia de hibernação através do mecanismo *deep sleep* (ver Tabela 1, D6). No entanto, como se trata de uma rede WiFi, numa topologia em malha, é necessário coordenar o acordar e adormecimento dos nós para que sejam aproveitados todos os benefícios desta topologia. Assim, em caso de falha de comunicação com o nó ponte, o nó sensorial toma iniciativa de adormecer ao fim de um determinado número de tentativas de comunicação, por outro lado, em caso de sucesso o nó adormece ao receber o sinal de *sleep* enviado pelo nó ponte (ver Tabela 1, D4). O fluxograma representado na Figura 11 descreve o funcionamento do nó sensorial.

Nó Ponte

Na arquitetura acima descrita e ilustrada na Figura 9 estão representados dois nós ponte. O nó ponte tem um papel muito importante na rede, pois é responsável por receber os dados recolhidos pelos nós sensoriais e remetê-los para o nó *gateway*. Para prevenir falhas decorrentes da indisponibilidade de um nó ponte (e.g., avaria), é essencial que a rede possua um mecanismo de tolerância a falhas. Para tal, adotou-se o uso de dois nós ponte, estando apenas um deles ativo. Caso exista uma falha no nó ponte ativo, o outro assume o seu lugar de forma automática e transparente, sem afetar o funcionamento dos restantes nós da rede. O mecanismo de redundância desenvolvido tem um princípio de funcionamento inspirado no protocolo VRRP (*Virtual Router Redundancy Protocol*)

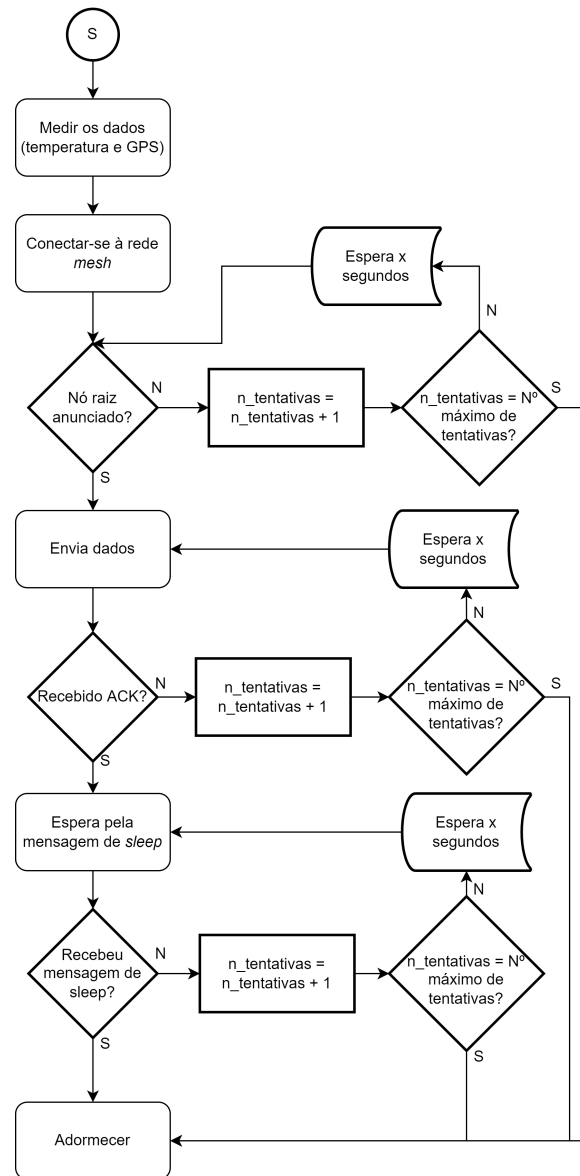


Figura 11. Fluxograma do nó sensorial.

[11] conforme se pode verificar na Figura 12. Através de um processo de seleção é determinado um nó raiz e o outro nó fica como *backup* (ver Tabela 1, D2). O processo de seleção consiste em cada nó ponte gerar um número aleatório e enviar uma mensagem com o número aleatório gerado, ao receber esta mensagem deve determinar se o número aleatório gerado é maior ou menor do que o recebido, se for menor então torna-se um nó *backup*, se o número for maior então toma o lugar de nó raiz. Em caso de empate (ou seja, os números aleatórios são iguais), é utilizado o ID do nó como fator de desempate, aplicando a mesma lógica do número aleatório, se o ID do nó for maior que o ID recebido então torna-se um nó raiz. O fluxograma apresentado na Figura 12 descreve o funcionamento inicial do nó ponte, onde é iniciado o algoritmo de VRRP.

O nó raiz recebe os dados dos outros nós sensoriais, co-

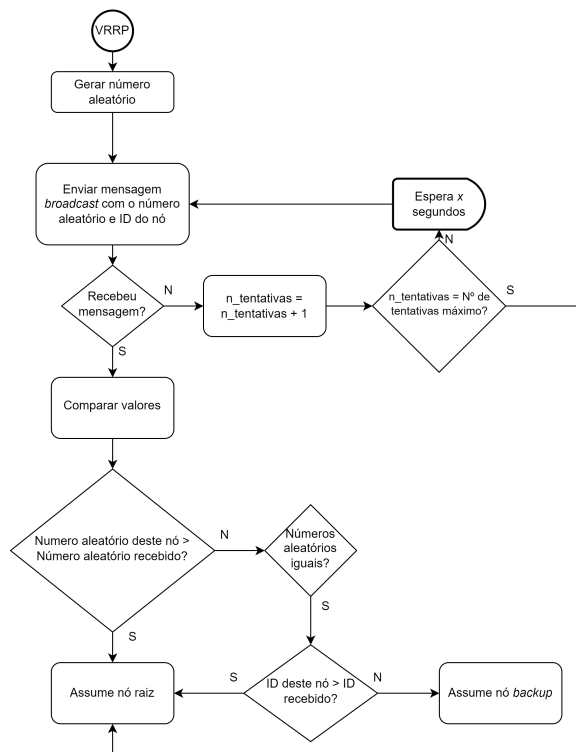


Figura 12. Fluxograma do algoritmo de eleição do ponte.

manda o adormecimento dos nós sensoriais e reenvia os dados recebidos para o nó *gateway*, ao mesmo tempo que envia sinais de *heartbeat* ao nó de raiz *backup*. O nó *backup* tem como função “escutar” os sinais de *heartbeat* e caso não receba um determinado número de sinais de *heartbeat*, então toma o lugar do nó raiz. A Tabela 2 descreve o algoritmo de funcionamento de cada tipo (nó raiz ou nó *backup*) que cada ponte pode desempenhar.

Nó Gateway

O nó *gateway* recebe os dados enviados através da rede LoRa pelo nó ponte. De seguida, através da rede WiFi com acesso à Internet, envia os mesmos para uma API através do uso do método POST do protocolo HTTP. Uma API é um mecanismo que permite que dois componentes de *software* comuniquem entre si, por outras palavras, permite que o nó *gateway* comunique os dados ao servidor web. O método HTTP POST é um pedido HTTP que permite enviar dados para um servidor através do campo *body*. O fluxograma presente na Figura 13 descreve o funcionamento do nó *gateway*. O nó *gateway* é um elemento fundamental da arquitetura, e um *single point of failure*, se existir apenas uma ligação à *internet* para envio dos dados. Em caso de falha, os dados recebidos através do nó ponte (e lidos pelos nós sensoriais) serão descartados, até que a ligação à *internet* seja estabelecida.

Cloud (API, Base de dados e Servidor Web)

A *cloud*, neste contexto, refere-se ao conjunto de serviços disponibilizados que tornam possível a receção, armazenamento e visualização dos dados recolhidos pela rede. Foi utilizado

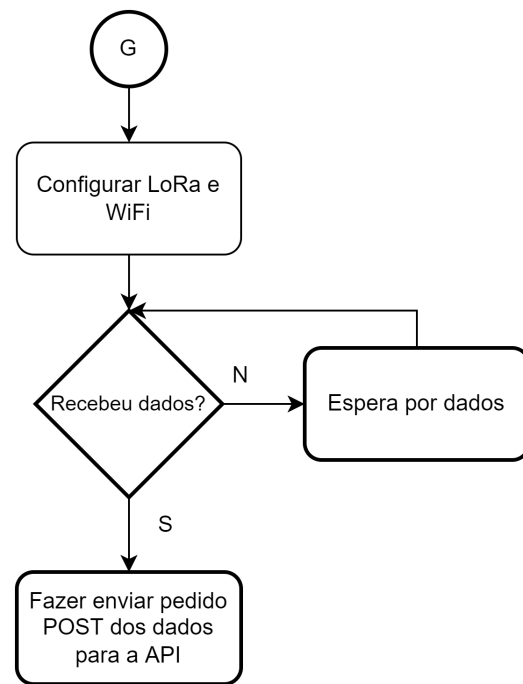


Figura 13. Fluxograma do nó *gateway*.

o *software* XAMPP que permite a execução de *scripts* PHP, a utilização de base de dados relacional e um servidor *web* [41] para visualização dos dados. O *script* PHP é uma API para receção dos dados enviados pelo nó *gateway* e posterior armazenamento. O *script* recebe, valida e separa os dados que são recebidos em formato JSON (*JavaScript Object Notation*) [42] em variáveis, para que o valor destas possa ser armazenado na base de dados relacional. O servidor *web* disponibiliza o site para visualização dos dados recolhidos pela rede que estão armazenados na base de dados. A Figura 14 mostra o funcionamento da *cloud*.

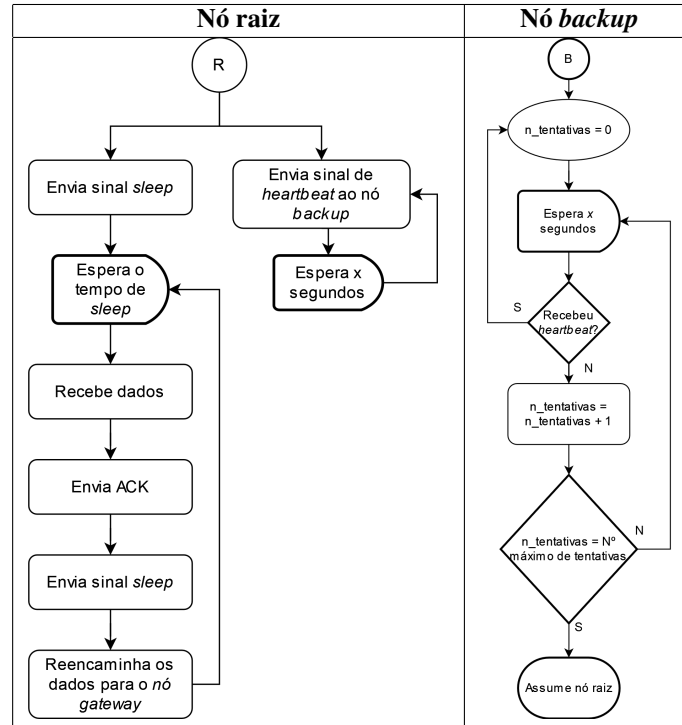


Figura 14. Diagrama do funcionamento da *cloud* (API, Base de Dados e Servidor Web).

4. Validação do Protótipo

Nesta secção apresentam-se os testes realizados ao protótipo. Procura-se validar o comportamento dos diferentes tipos de nós, a eficácia e eficiência da arquitetura global proposta, em cenários com e sem falhas de comunicação. A *testbed* utilizada para estes testes é composta por dois nós sensoriais, dois nós ponte e um nó *gateway*. Para demonstrar o funcionamento dos nós apresentam-se outputs recolhidos com o programa PuTTY [43]. O PuTTY é um programa que possui uma interface gráfica para a implementação das ligações SSH e Telnet, que também permite o monitoramento das portas

Tabela 2. Fluxogramas dos tipos de nós ponte.



serial. Para testar o funcionamento da API recorreu-se ao programa Postman [44]. O website foi testado num navegador web.

Nó Sensorial

O nó sensorial conforme descrito acima, tem como função a leitura dos dados e envio dos mesmos. O seu funcionamento tem por base 4 estados conforme se verifica na Figura 15. O primeiro estado é a leitura dos dados sensoriais, no caso temperatura e humidade. No segundo estado, o nó sensorial espera que o nó raiz se anuncie e nesse momento envia a mensagem com os dados recolhidos. Após enviar a mensagem, no terceiro estado, espera pelo ACK. Se não receber, vai reenviando a mensagem, nos testes foram usadas dez tentativas. No quarto e último estado, o nó sensorial espera pelo sinal de sleep do nó raiz. Este estado serve para sincronizar os nós de maneira que estes adormeçam e acordem ao mesmo tempo, para que se possa formar a rede em malha.

A abordagem de armazenar dados nos nós sensoriais é passível de causar problemas. Para armazenar dados tem que se considerar o limite da memória disponível, pois caso esta seja excedida o dispositivo reinicia automaticamente, perdendo os dados e o estado em que encontrava, o que iria resultar numa dessincronização com o resto dos nós. Outra opção seria armazenar apenas os dados mais recentes, no entanto seria necessário descartar os dados mais antigos, dando prioridade aos mais recentes, o que causaria sempre uma perda de dados. Pelas razões enunciadas, optou-se por não armazenar os dados e enviar apenas a leitura mais recente.

É importante notar que cada estado tem um número má-

ximo de tentativas (nos testes foram usadas dez tentativas). Isto é, caso não seja satisfeita a condição para a transição entre estados o nó ficará a executar esse mesmo estado durante um número determinado de vezes. Ao fim do número de tentativas estipulado se o nó não transitar de estado, então o nó irá adormecer por iniciativa própria (ver Tabela 1, D5).



Figura 15. Output do funcionamento do nó sensorial.

Nó Ponte

O nó ponte é um nó responsável por agregar os dados enviados pelos nós sensoriais e reencaminhá-los para o nó gateway. Ao mesmo tempo, este nó, envia o sinal sleep para sincronizar os nós sensoriais permitindo assim que estes adormeçam e acor-

dem ao mesmo tempo. O nó ponte é um elemento essencial ao funcionamento do protótipo, pelo que, a sua disponibilidade é fulcral. Como tal, foi implementado um algoritmo de negociação semelhante ao VRRP que permite que em caso de falha de um dos nós ponte, este seja substituído pelo outro. A Figura 16 apresentam o funcionamento do algoritmo VRRP em caso de falha de comunicação entre os nós ponte. As Figuras 17 e 18 apresentam o funcionamento do algoritmo VRRP em caso de sucesso, na vista nó backup e nó raiz, respetivamente. A Figura 19 apresenta o funcionamento do nó raiz.

```

STARTUP: init(): 0
STARTUP: AP tcp server established on port 5555
LoRa Sender
LoRa Initializing OK!
[Negotiation] Start!
--> startHere: New Connection, nodeId = 2785280700
Changed connections
Changed connections
[Negotiation] Start!
[Negotiation] {"mode":"negotiation","rNumber":8158,"id":2785280882}
[Negotiation] Waiting to receive negotiation message...
[Negotiation] Waiting to receive negotiation message...
[Negotiation] Waiting to receive negotiation message...
[Negotiation] Waiting to receive negotiation message...
[Negotiation] Waiting to receive negotiation message...
Changed connections
[Negotiation] Waiting to receive negotiation message...
[Negotiation] No negotiation message received, assuming root node
    
```

Configuração da biblioteca painlessMesh e da rede LoRa

Início do processo de negociação

Envio dos números aleatórios gerados e o id para negociação

Tentativa de comunicação com o outro nó ponte

Falha na comunicação e eleição do nó raiz

Figura 16. Funcionamento do algoritmo VRRP no nó ponte em caso de falha de comunicação.

```

STARTUP: init(): 0
STARTUP: AP tcp server established on port 5555
(Negotiation) Start!
--> startHere: New Connection, nodeId = 2785280721
Changed connections
Adjusted time 5083623. Offset = 2230447
Adjusted time 5259752. Offset = 1825
Changed connections
--> startHere: New Connection, nodeId = 576125939
(Negotiation) Start!
[Negotiation] {"mode":"negotiation","rNumber":8158,"id":2785280882}
startHere: Received from 576125939 msg="H
startHere: Received from 576125939 msg="{"mode":"sleep","time":60}
Changed connections
Changed connections
--> startHere: New Connection, nodeId = 2785280700
[Negotiation] Waiting to receive negotiation message...
startHere: Received from 576125939 msg="{"mode":"negotiation","rNumber":9065,"id":576125939}
[Negotiation] Assuming Backup Node!
startHere: Received from 576125939 msg="H
Changed connections
startHere: Received from 576125939 msg="H
startHere: Received from 576125939 msg="H
    
```

Configuração da biblioteca painlessMesh

Início da negociação

Valores gerados e mensagem enviada

Receção da mensagem de negociação e eleição do nó backup

Receção dos heartbeats

Figura 17. Funcionamento do algoritmo VRRP no nó ponte e eleição do nó backup.

```

STARTUP: init(): 0
STARTUP: AP tcp server established on port 5555
[Negotiation] Start!
--> startHere: New Connection, nodeId = 2785280882
Changed connections
Adjusted time 5761049. Offset = 3409672
Adjusted time 5966303. Offset = -4956
startHere: Received from 2785280882 msg="{"mode":"negotiation","rNumber":8158,"id":2785280882}
[Negotiation] Assuming Root Node!
Preparing Sleep signal
Sending message: {"mode":"sleep","time":60}
Sent nodes to sleep...zZz...
    
```

Configuração da biblioteca painlessMesh

Início da negociação

Processo de negociação o e eleição do nó raiz

Processo de sincronização dos nós sensoriais

Figura 18. Funcionamento do algoritmo VRRP no nó ponte e eleição do nó raiz.

```

[Root] Waiting for nodes to wake up...
Changed connections
--> startHere: New Connection, nodeId = 2785280721
Adjusted time 3924624. Offset = -9776986
Adjusted time 4028854. Offset = -6375
Adjusted time 4243014. Offset = 124
startHere: Received from 2785280721 msg="{"modeID":2785280721,"temperature":13,"humidity":23}
[Root] Sent message ACK! [Root] Waiting for data...
startHere: Received from 2785280700 msg="{"modeID":2785280700,"temperature":13,"humidity":23}
[Root] Sent message ACK! [Root] Waiting for data...
Preparing Sleep signal
Sending message: {"mode":"sleep","time":60}
Sent nodes to sleep...zZz...
Changed connections
[Root] Sending message: {"data":{"modeID":2785280721,"temperature":13,"humidity":23},"modeID":2785280700,"temperature":13,"humidity":23}}
    
```

Espera pelos nós sensoriais

Receção dos dados e envio dos ACKs

Envio do sinal sleep

Envio dos dados para o nó gateway através da rede LoRa

Figura 19. Funcionamento do nó ponte (nó raiz).

Nó Gateway

O nó gateway é o dispositivo que possui ligação à internet através de WiFi e conexão LoRa com os nós ponte (raiz e backup). Este nó reencaminha os dados recebido através da rede LoRa para uma API, que armazena os dados numa base de dados. A Figura 20 demonstra os outputs do funcionamento normal do nó gateway.

```

Connected! IP address: 192.168.0.16
LoRa Receiver Test
LoRa Initializing OK!
Received packet: {"data":{"modeID":2785280700,"temperature":25,"humidity":38}} [HTTP] begin...
[HTTP] POST...
[HTTP] POST... code: 201
Received packet: {"data":{"modeID":2785280721,"temperature":13,"humidity":42},"modeID":2785280700,"temperature":25,"humidity":38}} [HTTP] begin...
[HTTP] POST... code: 201
    
```

Conexão com a rede WiFi

Conexão com a rede LoRa

Resultado do pedido HTTP POST (201 = Created)

Receção e pedido HTTP POST com os dados para a API

Figura 20. Funcionamento do nó gateway.

Cloud (Website, API e Base de Dados)

A cloud tem como função demonstrar os dados recebidos através da API e armazenados na base de dados. A Figura 21 apresenta a visualização dos dados recebidos pelos nós sensoriais.

Projeto II - Vitor Aleluia

Node Identificar	Temperature	Humidity	Date
2785280721	13	27	2022-09-08 19:34:49
2785280700	25	38	2022-09-08 19:34:49
2785280700	25	38	2022-09-08 19:33:19
2785280721	23	57	2022-09-08 19:33:19
2785280700	25	38	2022-09-08 19:31:40
2785280721	3	42	2022-09-08 19:31:39
2785280700	13	23	2022-09-08 19:29:59
2785280721	3	42	2022-09-08 19:29:59
2785280700	25	38	2022-09-08 19:28:29
2785280721	3	42	2022-09-08 19:28:29
2785280700	25	38	2022-09-08 19:26:50

Figura 21. Website com os dados enviados pelos nós sensoriais.

5. Conclusões e Trabalho Futuro

O presente trabalho resulta de um projeto de investigação para proposta de uma solução para monitorização remota de animais de gado em larga escala. O trabalho apresentado teve como principal objetivo o desenho, implementação e avaliação de um protótipo funcional para demonstrar o conceito.

Utilizando dispositivos IoT, D1 Mini Pro e TTGO (com microcontroladores ESP8266 e ESP32), foi criada uma solução de rede em malha (gerida pela biblioteca `painlessMesh`) conectada com tecnologias de comunicação WiFi e LoRa. Para gestão da conectividade dos dispositivos, transportados pelos animais, com a rede em malha, foram desenvolvidos e avaliados vários algoritmos que garantiram, não só essa conectividade, mas também a economia dos gastos energéticos das baterias dos próprios dispositivos. Esta solução permitiu demonstrar a efetiva recolha de dados dos animais (através de sensoriais) e o seu envio para uma base de dados remota. O acesso a esta informação foi depois possibilitado, através de uma aplicação web, para consulta e análise. Os resultados obtidos confirmam que a abordagem proposta é capaz de dar resposta à necessidade de uma rede para monitorização que comporte grandes volumes de dados, que cresça funcionalmente e aumente a disponibilidade da rede com o aumento dos dispositivos.

Para trabalho futuro, pretende-se realizar testes de avaliação de desempenho no protótipo, para analisar o impacto causado pelo número de diferentes tipos de nós e de diferentes volumes de dados recolhidos, na taxa de entrega de mensagens fim-a-fim, e na autonomia e eficiência energética dos nós. Pretende-se também avançar com o desenvolvimento do colar, para teste e demonstração da solução aqui apresentada com animais de gado em ambiente real. Para além disso, pretende-se melhorar o frontend (visualização de dados) para que seja possível categorizar os dados por dispositivos, geração de gráficos e envio de alertas para o utilizador/agricultor.

Contribuições dos autores

Conceitualização: Vitor M. T. Aleluia, Vasco N. G. J. Soares, João M. L. P. Caldeira.

Metodologia: Vitor M. T. Aleluia, Vasco N. G. J. Soares, João M. L. P. Caldeira.

Análise Formal: Vitor M. T. Aleluia, Vasco N. G. J. Soares, João M. L. P. Caldeira.

Validação: Vasco N. G. J. Soares, João M. L. P. Caldeira, Pedro D. Gaspar.

Investigação: Vitor M. T. Aleluia.

Recursos: Vasco N. G. J. Soares, João M. L. P. Caldeira, Pedro D. Gaspar.

Escrita do original: Vitor M. T. Aleluia.

Revisão e edição: Vasco N. G. J. Soares, João M. L. P. Caldeira, Pedro D. Gaspar.

Supervisão: Vasco N. G. J. Soares, João M. L. P. Caldeira, Pedro D. Gaspar.

Administração de Projeto: Vasco N. G. J. Soares, João M. L. P. Caldeira, Pedro D. Gaspar.

References

- [1] FROST, A. R. et al. A review of livestock monitoring and the need for integrated systems. *Computers and Electronics in Agriculture*, Elsevier, Netherlands, v. 17, n. 2, p. 139–159, May 1997.
- [2] ABDULLAHI, U. S.; ORISEKEH, K.; NYABAM, M. A. Exploiting IoT and LoRaWAN technologies for effective livestock monitoring in Nigeria. *The Arid Zone Journal of Engineering, Technology and Environment*, Maiduguri, Nigeria, v. 15, n. SPi2, p. 146–159, June 2019.
- [3] GERMANI, L. et al. An IoT architecture for continuous livestock monitoring using lora LPWAN. *Electronics*, MDPI AG, Switzerland, v. 8, n. 12, p. 1–28, December 2019.
- [4] What is Arduino? Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 14 de janeiro de 2023.
- [5] RASPBERRY Pi Foundation – About us. Disponível em: <<https://www.raspberrypi.org/about/>>. Acesso em: 14 de janeiro de 2023.
- [6] WHAT is LoRaWAN® Specification - LoRa Alliance®. Disponível em: <<https://lora-alliance.org/about-lorawan/>>. Acesso em: 14 de janeiro de 2023.
- [7] SIGFOX. Disponível em: <<https://www.sigfox.com/en/what-sigfox/technology>>. Acesso em: 14 de janeiro de 2023.
- [8] SINHA, R. S.; WEI, Y.; HWANG, S. H. A survey on lpwa technology: Lora and nb-iot. *ICT Express*, Elsevier, Beijing, China, v. 3, n. 1, p. 14–21, 3 2017.
- [9] HAXHIBEQIRI, J. et al. A survey of lorawan for iot: From technology to application. *Sensors*, Multidisciplinary Digital Publishing Institute, Basel, Switzerland, v. 18, n. 11, p. 1–38, November 2018.
- [10] ALELUIA, V. M. T. et al. Livestock monitoring: Approaches, challenges and opportunities. *International Journal of Engineering and Advanced Technology*, Bhopal, India, v. 11, n. 4, p. 67–76, April 2022.
- [11] IMPLEMENTING VRRP - Cisco. Disponível em: <https://www.cisco.com/c/en/us/td/docs/routers/crs/software/crs_r4-0/addr_serv/configuration/guide/ic40crs1book_chapter10.html>. Acesso em: 14 de janeiro de 2023.
- [12] REIGONES, A. R.; GASPAR, P. D. Real-time vital signs monitoring system towards livestock health furtherance. In: *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*. New York, USA: IEEE, 2021. p. 753–758.
- [13] BITALINO. Disponível em: <<https://bitalino.com/products/r-iot-2>>.

- [14] WHAT Is Wi-Fi? - Definition and Types - Cisco. Disponível em: <<https://www.cisco.com/c/en/us/products/wireless/what-is-wifi.html>>. Acesso em: 14 de janeiro de 2023.
- [15] COWLAR. Disponível em: <<https://www.dairy.cowlar.com>>. Acesso em: 14 de janeiro de 2023.
- [16] ENVIRA IOT. Disponível em: <<https://enviraiot.com/animal-health-monitoring-farms/>>. Acesso em: 14 de janeiro de 2023.
- [17] NFC Group. Disponível em: <<https://www.tracks360.com/about/>>. Acesso em: 14 de janeiro de 2023.
- [18] MAROTO-MOLINA, F. et al. A low-cost iot-based system to monitor the location of a whole herd. *Sensors*, Basel, Switzerland, v. 19, n. 10, p. 1–15, May 2019.
- [19] GPS.GOV: GPS Overview. Disponível em: <<https://www.gps.gov/systems/gps/>>.
- [20] BLUETOOTH Low Energy – Revolutionizing Connectivity for the IoT | Bluetooth® Technology Website. Disponível em: <<https://www.bluetooth.com/bluetooth-resources/bluetooth-low-energy-revolutionizing-the-connectivity-for-iot-2>>. Acesso em: 14 de janeiro de 2023.
- [21] GPS ear tags | mOOvement. Disponível em: <<https://www.moovement.com.au/>>. Acesso em: 14 de janeiro de 2023.
- [22] MOOVEMENT GPS Cattle Tags, Track and Trace Animals With Your Phone - 4Tags.com.au. Disponível em: <<https://4tags.com.au/movement-gps-cattletags/>>.
- [23] PARK, J. K.; PARK, E. Y. Animal monitoring scheme in smart farm using cloud-based system. *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, Pathum Thani, Thailand, v. 15, n. 1, p. 24–33, April 2021.
- [24] JUANG, P. et al. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. *ACM SIGPLAN Notices*, New York, USA, v. 37, n. 10, p. 96–107, October 2002.
- [25] MARTONOSI, M. *The Princeton ZebraNet Project: Sensor Networks for Wildlife Tracking*. Disponível em: <<http://mpsoc-forum.org/archive/2004/slides/Martonosi.pdf>>. Acesso em: 14 de janeiro de 2023.
- [26] LORA and LoRaWAN: Technical overview | DEVELOPER PORTAL. Disponível em: <<https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan>>. Acesso em: 14 de janeiro de 2023.
- [27] ALLIANCE, W.-F. *Securing Wi-Fi Wireless Networks with Today's Technologies Wi-Fi Alliance*. 2003. Disponível em: <http://www.netsense.info/downloads/Whitepaper_Wi-Fi_Networks2-6-03.pdf>. Acesso em: 14 de janeiro de 2023.
- [28] CASAS, R. et al. Real-time extensive livestock monitoring using lpwan smart wearable and infrastructure. *Applied Sciences*, Basel, Switzerland, v. 11, n. 3, p. 1–18, January 2021.
- [29] AJISH, K. K.; HEGDE, S. Multicasting in wireless mesh networks: Challenges and opportunities. In: *Proceedings - 2009 International Conference on Information Management and Engineering, ICIME 2009*. New York, USA: IEEE, 2009. p. 514–518.
- [30] PAINLESSMESH. Disponível em: <<https://gitlab.com/painlessMesh/painlessMesh>>.
- [31] WHAT is an Application Programming Interface (API) | IBM. Disponível em: <<https://www.ibm.com/cloud/learn/api>>. Acesso em: 14 de janeiro de 2023.
- [32] D1 mini Pro. Disponível em: <https://www.wemos.cc/en/latest/d1/d1_mini_pro.html>.
- [33] FREQUENCY Plans by Country. Disponível em: <<https://www.thethingsnetwork.org/docs/lorawan/frequencies-by-country>>. Acesso em: 14 de janeiro de 2023.
- [34] BOBKOV, I. et al. Study of lora performance at 433 mhz and 868 mhz bands inside a multistory building. In: *Moscow Workshop on Electronic and Networking Technologies, MWENT 2020 - Proceedings*. New York, USA: Institute of Electrical and Electronics Engineers Inc., 2020. p. 1–6.
- [35] TTGO LoRa32-OLED V1 — PlatformIO latest documentation. Disponível em: <<https://docs.platformio.org/en/latest/boards/espressif32/ttgo-lora32-v1.html#ttgo-lora32-oled-v1>>. Acesso em: 14 de janeiro de 2023.
- [36] SANTOS, S. *ESP32 vs ESP8266 - Pros and Cons - Maker Advisor*. Disponível em: <<https://makeradvisor.com/esp32-vs-esp8266/>>. Acesso em: 14 de janeiro de 2023.
- [37] RFM95W feature the LoRaTM long range modell Hoperf. Disponível em: <<https://www.hoperf.com/modules/lora/RFM95.html>>. Acesso em: 14 de janeiro de 2023.
- [38] SANDEEPMISTRY/ARDUINO-LORA: An Arduino library for sending and receiving data using LoRa radios. Disponível em: <<https://github.com/sandeepmistry/arduino-LoRa>>. Acesso em: 14 de janeiro de 2023.
- [39] AMCEWEN/HTTPCLIENT: Arduino HTTP library. Disponível em: <<https://github.com/amcewen/HttpClient>>. Acesso em: 14 de janeiro de 2023.
- [40] A professional collaborative platform for embedded development · PlatformIO. Disponível em: <<https://platformio.org/>>. Acesso em: 14 de janeiro de 2023.
- [41] WHAT is a web server? | MDN Web Docs. Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server>. Acesso em: 14 de janeiro de 2023.

[42] JSON. Disponível em:<<https://www.json.org/json-en.html>>. Acesso em: 14 de janeiro de 2023.

[43] PuTTY: a free SSH and Telnet client. Disponível em:<<https://www.chiark.greenend.org.uk/~sgtatham/putty/>>.

Acesso em: 14 de janeiro de 2023.

[44] Postman API Platform. Disponível em:<<https://www.postman.com/>>. Acesso em: 14 de janeiro de 2023.