



UNIVERSIDADE ESTADUAL PAULISTA

“JÚLIO DE MESQUITA FILHO”

Campus de Sorocaba - SP

Rodrigo de Souza Jacomini

**SISTEMA EMBARCADO APLICADO A AUTOMAÇÃO NA
ALIMENTAÇÃO DE PETS: ML E IoT**

Sorocaba

2024

Rodrigo de Souza Jacomini

**SISTEMA EMBARCADO APLICADO A AUTOMAÇÃO NA
ALIMENTAÇÃO DE PETS: ML E IoT**

Dissertação apresentada como parte dos requisitos para obtenção de título de Mestre em Engenharia Elétrica, junto ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Estadual Paulista “Júlio de Mesquita Filho”.

Área de concentração: Automação.

Orientador: Prof. Dr. Ivando Severino Diniz

Sorocaba

2024



J17s

Jacomini, Rodrigo de Souza

Sistemas embarcado aplicado a automação na alimentação de pets:
ML e IoT / Rodrigo de Souza Jacomini. -- Sorocaba, 2024
117 p.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP),
Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Ivando Severino Diniz

1. Alimentação dos animais. 2. Automação residencial. 3.
Inteligência artificial. 4. Indústria animal. 5. Circuitos
eletrônicos-Projetos. I. Título.

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Sistema Embarcado para Controle de Alimentação de Pets Baseado em Aprendizado de Máquina e IoT

AUTOR: RODRIGO DE SOUZA JACOMINI

ORIENTADOR: IVANDO SEVERINO DINIZ

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica, área: Automação pela Comissão Examinadora:

Prof. Dr. IVANDO SEVERINO DINIZ (Participação Presencial)
Departamento de Engenharia de Controle e Automação / Instituto de Ciência e Tecnologia / UNESP / Sorocaba

Prof. Dr. EDUARDO VERRI LIBERADO (Participação Presencial)
Departamento de Engenharia de Controle e Automação / Instituto de Ciência e Tecnologia - UNESP - Câmpus de Sorocaba

Prof. Dr. SERGIO SHIMURA (Participação Presencial)
Departamento de Indústria / Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus Sorocaba

Sorocaba, 27 de novembro de 2024

Documento assinado digitalmente
 **IVANDO SEVERINO DINIZ**
Data: 29/11/2024 08:26:37-0300
Verifique em <https://validar.iti.gov.br>

 Assinado de forma digital
por Eduardo Verri Liberado
Dados: 2024.11.29 09:43:27
-03'00'

Documento assinado digitalmente
 **SERGIO SHIMURA**
Data: 29/11/2024 20:09:34-0300
Verifique em <https://validar.iti.gov.br>

Dedico esse trabalho a toda minha família e
principalmente a minha netinha Lili.

AGRADECIMENTOS

Agradeço, em primeiro lugar, ao Divino Mestre Gabriel, fonte de sabedoria e força, e aos meus familiares, pelo amor e apoio incondicional. Registro minha gratidão ao Professor Dr. Ivando Severino Diniz, pela orientação e dedicação ao longo desta jornada acadêmica. Agradeço também aos funcionários, professores e colegas da UNESP Sorocaba – SP, que contribuíram para o enriquecimento desta experiência. Quero expressar um agradecimento especial ao meu irmão, Ricardo de Souza Jacomini que é Mestre, Doutor e Ph.D., por todo o apoio, paciência e valiosas orientações durante a elaboração desta dissertação. Sua vasta experiência acadêmica foi essencial para me guiar nesse processo, e seu incentivo constante me mostrou que nunca estive sozinho nesta caminhada. Sou profundamente grato por todo o aprendizado compartilhado e por ser uma inspiração contínua em minha vida.

“A paz é a única forma de nos sentirmos realmente humanos.”
Albert Einstein

RESUMO

O Brasil se destaca globalmente na indústria de animais de estimação, com significativo faturamento e uma vasta população de aproximadamente 160 milhões de animais, incluindo 67,8 milhões de cachorros e 33,6 milhões de gatos. No entanto, a ausência dos tutores durante viagens ou períodos de trabalho pode comprometer o cuidado adequado com os animais. Nesse contexto, a tecnologia, especialmente a Internet das Coisas (IoT), surge como uma solução para facilitar o cuidado com os animais de estimação. Muitas soluções disponíveis para monitoramento e alimentação de animais apresentam limitações, como curto alcance de identificação e alto consumo de energia. Assim, há uma demanda por dispositivos mais avançados e eficientes capazes de fornecer cuidados personalizados aos animais e lidar com falhas de energia. Esta dissertação apresenta o desenvolvimento de um sistema inteligente que combina hardware e software para identificar animais, lidar com falhas de energia elétrica e fornecer alimentação personalizada e garantir o bem-estar dos animais de estimação, mesmo na ausência do tutor e em situações de falta de energia. No presente estudo, foi desenvolvido um dispositivo capaz de alimentar e oferecer água a diferentes animais de estimação a partir de treinamentos realizados por meio dos algoritmos FOMO e YOLO. Além disso, foi criado um nobreak inédito para a manutenção da funcionalidade do dispositivo mesmo na ausência de energia elétrica. Como resultados, foi possível observar que o dispositivo desenvolvido foi capaz de distinguir cães e gatos com uma sensibilidade e especificidade adequadas, no entanto, o algoritmo de classificação YOLO (83,3%) se saiu melhor que o FOMO (75%). Além disso, observou-se que, na ausência de energia elétrica, o sistema continuou funcionando por 60 horas. Na ausência de energia elétrica, o dispositivo ainda foi capaz de enviar dados sobre alimentação e consumo de água pelo animal de estimação para um cartão *SD Card* e para nuvem. Testes realizados com um cão em uma casa mostraram que o sistema foi facilmente utilizável. Os resultados obtidos destacam o desenvolvimento bem-sucedido de um dispositivo inteligente que atende às necessidades dos tutores de animais de estimação, proporcionando tranquilidade e bem-estar tanto para os animais quanto para seus cuidadores. O dispositivo desenvolvido demonstrou eficiência na identificação e alimentação dos animais, além de garantir a operação contínua, mesmo em situações de falta de energia elétrica. Análises futuras visarão testar o dispositivo em um ambiente real.

Palavras-chave: Internet das Coisas (IoT); Inteligência Artificial (IA); Dispositivo inteligente; Resiliência a falta de energia; Alimentação personalizada; Cuidados com animais de estimação; Automação residencial.

ABSTRACT

Brazil stands out globally in the pet industry, with significant revenue and a vast population of approximately 160 million pets, including 67.8 million dogs and 33.6 million cats. However, the absence of owners during travel or work periods can compromise proper pet care. In this context, technology, especially the Internet of Things (IoT), emerges as a solution to facilitate pet care. Many available solutions for monitoring and feeding pets have limitations such as short identification range and high energy consumption. Thus, there is a demand for more advanced and efficient devices capable of providing personalized care to pets and handling power failures. This dissertation presents the development of an intelligent system combining hardware and software to identify animals, manage power failures, provide personalized feeding, and ensure pet well-being even in the absence of owners and during power outages. The study developed a device capable of feeding and offering water to different pets based on training using FOMO and YOLO algorithms. Additionally, a novel uninterruptible power supply (UPS) was created to maintain device functionality during power outages. Results show that the developed device could distinguish between dogs and cats with adequate sensitivity and specificity, with the YOLO classification algorithm performing better (83.3%) than FOMO (75%). Furthermore, in the absence of electricity, the system continued to operate for 60 hours and was able to send feeding and water consumption data by the pet to an SD card and cloud storage. Tests conducted with a dog in a home environment demonstrated the system's ease of use. Overall, the results highlight the successful development of an intelligent device meeting the needs of pet owners, providing peace of mind and well-being for both pets and their caregivers. The device demonstrated efficiency in animal identification and feeding, along with ensuring continuous operation even in power outage situations. Future analyses will focus on testing the device in a real-world environment.

Keywords: Internet of Things (IoT); Artificial intelligence (AI); Smart device; Power outage resilience; Personalized feeding; Pet care; Smart Homes.

LISTA DE FIGURAS

Figura 1 – Estrutura mecânica dividida em cinco partes	34
Figura 2 – Detalhes alimentador por gravidade.....	35
Figura 3 – Detalhes mecânicos interno do dosador de ração.....	35
Figura 4 – Detalhes da estrutura.....	36
Figura 5 – Esquema elétrico do alimentador	38
Figura 6 – Detalhes da montagem dos componentes elétricos	39
Figura 7 – Arquitetura do sistema	40
Figura 8 – Integração da comunicação entre todas as placas.....	41
Figura 9 – Diagrama em bloco do Hardware do Nobreak Inteligente	43
Figura 10 – Vista em 3D da PCI do nobreak no software Easyeda	44
Figura 11 – Vista em 2D do nobrek com todos os detalhes	45
Figura 12 – Gravador USBASP	46
Figura 13 – Esquema eletrônico do gravador USBASP	47
Figura 14 – Sensor 220ac.....	48
Figura 15 – Esquema Eletrônico do Sensor DC.....	49
Figura 16 – Sequencia de cálculo do sensor DC	49
Figura 17 – Esquema do circuito da célula de carga.....	50
Figura 18 – Estrutura física da balança da ração	52
Figura 19 – Peso padrão de 200 gramas - calibrar a balança	53
Figura 20 – Firmware de calibração da balança.....	54
Figura 21 – DF500 modelo de bateria estacionária.....	55
Figura 22 – Processador ESP32CAM	57
Figura 23 – Dolly a cachorra utilizada no treinamento.....	58
Figura 24 – Passos da <i>Edge Impulse</i>	58
Figura 25 – Captura das fotos na plataforma <i>Edge Impulse</i>	59
Figura 26 – Inclusão do <i>label</i> na plataforma <i>Edge Impulse</i>	60
Figura 27 – <i>Impulse Design</i> na plataforma <i>Edge Impulse</i>	61
Figura 28 – Parametrização da rede neural - Plataforma <i>Edge Impulse</i>	62
Figura 29 – Resultado da rede neural - <i>Edge Impulse</i>	63
Figura 30 – Escolha de qual microcontrolador irá rodar o algoritmo.....	64
Figura 31 – Gerenciamento e inclusão da biblioteca.....	65
Figura 32 – Algoritmo de classificação para esp32_câmera	66
Figura 33 – Projeto esp32_câmera aberto na IDE Arduino	67
Figura 34 – Escolha da placa AI Thinker ESP32-CAM.....	68
Figura 35 – Esquema eletrônico do gravador da ESP32-CAM.....	69
Figura 36 – Base da ESP32-CAM.....	70
Figura 37 – Pinagem da ESP32-WROOM-32	71
Figura 38 – PCI do <i>Datalogger</i>	72
Figura 39 – Esquema de ligação de todas PCIs	74
Figura 40 – Divisão das tarefas nos dois core da ESP32.....	76
Figura 41 – Lógica de registro na nuvem e gravação no cartão SD	77
Figura 42 – Lógica de controle da ração	78
Figura 43 – Lógicas de controle da água e da luz.....	79
Figura 44 – Cabeçalho, configuração do servidor <i>Ubidots</i> e Wi-fi	80
Figura 45 – Função callback.....	81
Figura 46 – Função setup	81
Figura 47 – Função principal.....	82
Figura 48 – Esquema eletrônico de teste do servidor <i>Ubidots</i>	83
Figura 49 – Hardware de teste do servidor <i>Ubidots</i>	83
Figura 50 – Conta gratuita no servidor MQTT <i>Ubidots</i>	84
Figura 51 – Dispositivo esp32C publicando no <i>Ubidots</i>	85
Figura 52 – Dashborad aberto no PC	86

Figura 53 – <i>Dashboard</i> pronto e aberto no celular	86
Figura 54 – Projeto IoT e fluxo dos dados publish e subscribe	87
Figura 55 – Dispositivo publicando 6 variáveis no <i>Ubidots</i>	89
Figura 56 – Variável botão virtual da ração	90
Figura 57 – Variável da boia física ligada a ESP32.....	90
Figura 58 – Variável da célula de carga da balança.....	90
Figura 59 – Variável da tensão DC da bateria	91
Figura 60 – Variável da tomada 127 V AC	91
Figura 61 – <i>Dashboard</i> completo do projeto da dissertação	92
Figura 62 – Classificação gato pela ESP32_CAM	96
Figura 63 – Classificação (Cachorro) pela ESP32_CAM	97
Figura 64 – Resultado do Sensor AC.....	98
Figura 65 – Resultado do Sensor DC	99
Figura 66 – Resultado da balança funcionando	100
Figura 67 – Resultado da autonomia de 60 horas da bateria.....	102
Figura 68 – <i>Dashboard</i> do Sistema funcionando pela bateria.....	103
Figura 69 – <i>Dashboard</i> do sistema AC e DC normalizados	104
Figura 70 – Arquivo interno do <i>Datalogger</i> gerado no cartão SD	105
Figura 71 – Resultado físico da dissertação do Mestrado	107
Figura 72 – Teste do equipamento real na casa com a cachorra.....	108

LISTA DE TABELAS

Tabela 1 – Comparação da dissertação com estado da arte	32
Tabela 2 – Consumo dos equipamentos eletroeletrônicos	56
Tabela 3 – Matriz de confusão do algoritmo YOLO	94
Tabela 4 – Matriz de Confusão: Treinamento do modelo aprendido usando o FOMO	94

LISTA DE SIGLAS

ABINPET	Associação Brasileira da Indústria de Produtos para Animais de Estimação
AC	<i>Alternating Current</i> (Corrente Alternada)
AD0	Analógico/Digital 0
AH	Amper Hora
APP	<i>Application</i> (Aplicativo)
AVR	Um Microcontrolador da empresa <i>Microchip</i>
AWS	<i>Amazon Web Services</i> – (Serviços Web Amazon)
BLE	<i>Bluetooth Low Energy</i> (Bluetooth de Baixa Energia)
CI	<i>Integrated Circuit</i> (Circuito Integrado)
DC	<i>Direct Current</i> (Corrente Contínua)
EEPROM	<i>Electrically Erasable Programmable Read Only Memory</i>
FC1	Fim-de-curso 1
FOMO	<i>Faster Objects, More Objects</i> (Objetos mais rápidos, mais objetos)
GPIO	<i>General-Purpose Input/Output</i> (Entrada/Saída de Propósito Geral)
IA	<i>Artificial Intelligence</i> (Inteligência Artificial)
I2C	<i>Inter-Integrated Circuit</i> (Circuito Inter-Integrado)
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i> (Internet das Coisas)
IR	<i>Infrared</i> – Infravermelho
I2S	<i>Integrated Inter-IC Sound</i> (Som Inter-IC Integrado)
LED	<i>Light Emitting Diode</i> (Diodo Emissor de Luz)
mA	Milésimo de Amper
MHz	Megahertz
ML	<i>Machine Learning</i> (Aprendizado de Máquina)
M2M	<i>Machine to Machine</i> (Máquina a Máquina)
MP	Mega Pixel
MQTT	<i>Message Queuing Telemetry Transport</i>
NTP	<i>Network Time Security</i> (Rede de Relógio Seguro)
PC	<i>Personal Computer</i> (Computador Pessoal)
PCB	<i>Printed Circuit Board</i>
PCI	Placa de Circuito Impresso
PVC	Poli Cloreto de Vinila
PSC	Placa do Sistema de Controle (<i>Datalogger</i>)
PWM	<i>Pulse-Width Modulation</i> (Modulação por Largura de Pulso)
RAM	<i>Random Access Memory</i> (Memória de Acesso Aleatório)

RFID	Identificador por Rádio Frequência
RMS	<i>Root Mean Square</i> (Raiz Média Quadrada)
RNAs	Redes neurais artificiais
ROM	<i>Read-Only Memory</i> (Memória Somente de Leitura)
RTC	<i>Real Time Clock</i> (Relógio Tempo Real)
RX	Recepção
SASA	Síndrome de Ansiedade de Separação em Animais
SD	<i>Secure Digital</i> (Digital Segura)
SIP	Sistema de Identificação de Pets (ESP32-CAM)
SMD	<i>Surface Mounted Device</i> (Componentes de Montagem de Superfície)
SMS	Short Message Service (Serviço de Mensagem Curta)
SPI	<i>Serial Peripheral Interface</i> (Interface Periférica Serial)
SRAM	<i>Static Random Access Memory</i> (Memória de Acesso Aleatório e Estático)
SSEI	Sistema de Suprimento de Energia Inteligentes (Placa do Nobreak)
TCP/IP	<i>Transfer Control Protocol / Internet Protocol</i>
TTL	<i>Transistor-Transistor Logic</i> (Lógica Transistor-Transistor)
TX	Transmissão
UART	<i>Universal Asynchronous Receiver</i>
UNESP	Universidade Estadual Paulista
USB	<i>Universal Serial Bus</i> (Barramento Serial Universal)
VAC	<i>Voltage Alternating Current</i> (Tensão Corrente Alternada)
VCC	<i>Direct Current Voltage</i> (Tensão Corrente Contínua)
VNC	<i>Virtual Network Computing</i> (Rede de Computador Virtual)
W	Watts
Wi-Fi	<i>Wireless Fidelity</i>
WPA	<i>Wi-Fi Protected Access</i> (Acesso Protegido por Wi-Fi)
WPS	<i>Wi-Fi Protected Setup</i> (Configuração Protegida por Wi-Fi)
YOLO	<i>You Only Look Once</i> (Você Só Olha uma Vez)

SUMÁRIO

1 INTRODUÇÃO	17
2 JUSTIFICATIVA	20
3 OBJETIVOS	21
3.1 OBJETIVO GERAL	21
3.2 OBJETIVOS ESPECÍFICOS	21
4 ORGANIZAÇÃO DA DISSERTAÇÃO	22
5 FUNDAMENTOS TEÓRICOS	23
5.1 IOT – INTERNET DAS COISAS	23
5.2 MQTT (<i>MESSAGE QUEUING TELEMETRY TRANSPORT</i>).....	24
5.3 AWS IOT – AMAZON.....	25
5.4 ML – MACHINE LEARNING (APRENDIZADO DE MÁQUINA).....	25
5.5 REDE NEURAL.....	26
5.6 <i>EDGE IMPULSE</i>	26
5.7 MECÂNICA – DIFERENTES MECANISMOS DE DOSAGEM DE RAÇÃO ..	27
6 REVISÃO DA LITERATURA.....	29
6.1 MERCADO PET NO BRASIL.....	29
6.2 UTILIZAÇÃO DE SISTEMAS INTELIGENTES PARA CUIDADOS DE PETS	29
7 MATERIAIS E MÉTODOS.....	33
7.1 ESTRUTURA MECÂNICA DO SISTEMA	33
7.2 ESQUEMA ELÉTRICO DO SISTEMA	36
7.3 ARQUITETURA DO SISTEMA	39
7.4 PROJETO NOBREAK INTELIGENTE	41
7.4.1 <i>Escopo do Nobreak Inteligente</i>	41
7.4.2 <i>Gravador USBASP</i>	46
7.4.3 <i>Sensor AC Voltage LM358</i>	47
7.4.4 <i>Sensor DC de 0 à 25Vdc</i>	49
7.4.5 <i>Sensor da Balança – Célula de Carga 3 kg – Amplificador hx711</i>	50
7.5 AUTONOMIA DOS SISTEMAS DE BATERIA E CARREGAMENTO	55
7.6 MICROCONTROLADOR ESP32CAM	56
7.7 PLATAFORMA <i>EDGE IMPULSE</i> – TREINAMENTO DA REDE NEURAL....	57
7.7.1 <i>Passo 1 - Edge Impulse (Coleta Imagens)</i>	59
7.7.2 <i>Passo 2 - Edge Impulse (Inclusão de Lables)</i>	60
7.7.3 <i>Passo 3 - Edge Impulse (Impulse Design)</i>	60
7.7.4 <i>Passo 4 - Edge Impulse (Deployment)</i>	63
7.7.5 <i>Passo 5 - Edge Impulse (Run Model)</i>	64
7.8 MICROCONTROLADOR ESP32-WROOM-32	70
7.9 SISTEMA IOT DO PROJETO	71
7.9.1 <i>IoT parte 1 - Hardware do sistema</i>	72
7.9.2 <i>IoT parte 2 - IDE de programação do sistema embarcado</i>	75
7.9.3 <i>IoT parte 3 – Ubidots – o Servidor</i>	84

7.9.4	<i>Ubidots aplicado ao projeto da dissertação</i>	86
8	RESULTADOS E DISCUSSÃO	93
8.1	RESULTADOS DO SISTEMA DE IDENTIFICAÇÃO DE PETS (SIP).....	93
8.2	RESULTADO INDIVIDUAL DO SENSOR AC.....	97
8.3	RESULTADO INDIVIDUAL DO SENSOR DC.....	98
8.4	RESULTADOS DA IMPLEMENTAÇÃO DA CÉLULA DE CARGA	99
8.5	RESULTADO DA AUTONOMIA DA BATERIA	102
8.6	SIMULAÇÃO SEM ENERGIA AC DA REDE (NOBREAK OPERANDO)	102
8.7	SIMULAÇÃO COM ENERGIA AC DA REDE.....	104
8.8	SIMULAÇÃO DO SISTEMA SEM INTERNET (DATALOGGER)	105
8.9	TESTE REAL EM CASA	106
9	CONCLUSÃO	109
10	PERSPECTIVAS FUTURAS	110
	REFERÊNCIAS	111

1 INTRODUÇÃO

O Brasil ocupa a terceira colocação mundial em faturamento com a indústria de animais de estimação, seguido da China e Estados Unidos (ABINPET, 2023). Em 2023, no Brasil, o faturamento total na indústria de animais de estimação foi igual a 47 bilhões, sendo dividido em 7% nos cuidados com os animais de estimação, 15% com veterinários e 78% com alimentação. Além disso, estima-se que a população de animais de estimação no Brasil é igual a 167,6 milhões, sendo 67,8 milhões só de cães (ABINPET, 2023). Assim, considerando uma população de 215 milhões de pessoas no país, é possível que aproximadamente 70% dela tenha ao menos um animal de estimação em casa (Valeri, 2023). Tais dados evidenciam a importância que brasileiros dão para seus animais de estimação. Entretanto, é comum que, em viagens a passeio ou trabalho, tutores precisem se ausentar por dias, afetando o nível de cuidado oferecido a seus animais domésticos.

Ao ficar sozinho, o animal está sujeito, em alguns casos, a apresentar problemas como 1) baixa frequência de alimentação ou 2) exagero na alimentação, que pode levar à obesidade (Lima et al., 2023). Além disso, os animais, quando sozinhos, podem apresentar a Síndrome de Ansiedade de Separação em Animais (SASA), que leva à vocalização excessiva, comportamentos compulsivos constantes fora de contexto, destruição de objetos, vômitos e depressão (SOARES; TELHADO; PAIXÃO, 2012). Esses fatores podem desencadear uma série de outros problemas para tutores e animais de estimação, como reclamações em ambientes de convívio compartilhado e mais gastos com a saúde do animal.

Com o avanço da tecnologia, são desenvolvidas constantemente ferramentas para auxiliar em necessidades diárias, trazendo mais conforto e praticidade para pessoas e seus animais de estimação (Lima et al., 2023; Devaki et al., 2023). Desta forma, essas tecnologias podem melhorar o aproveitamento do tempo em lazer e em família, pois nem sempre é possível levar o animal de estimação ao trabalho ou a passeios e viagens. Além disso, tecnologias que forneçam alimento e água para animais na ausência de tutores podem contribuir para uma melhor convivência e qualidade de vida do animal de estimação. Nesse contexto, a internet das coisas (IoT - *Internet of Things*) vêm auxiliando no controle e monitoramento de sensores em vários locais, inclusive nos ambientes de animais de estimação. Com o avanço dos sistemas digitais embarcados, sensores e comunicação de redes, a IoT vem sendo

aplicada na indústria, centros de pesquisas e no meio acadêmico (Santos et al., 2016). A tecnologia atual IoT tem a capacidade de medir, controlar e armazenar dados com a possibilidade de aplicação nas mais diversas áreas, como por exemplo na saúde, nas casas e cidades inteligentes (Molina, 2024).

Atualmente, existem dispositivos que visam monitorar e oferecer alimento e água para animais de estimação que ficam sozinhos ao longo do dia ou durante dias. O trabalho “Protótipo de um Alimentador para Cães com Identificador por Rádio Frequência” aplicou a tecnologia de reconhecimento de RFID (*Radio-frequency identification*) na coleira do cachorro. Nesse sistema, o animal é identificado por uma tag e o dispositivo oferece alimento e água ao animal assim que ele é reconhecido. No entanto, o sistema apresenta algumas limitações, como curta distância de leitura entre a tag e a antena do sensor. Com isso, há maior dificuldade no reconhecimento do animal de estimação. Além disso, o sistema depende integralmente da energia elétrica, de modo que, na ausência dela, o animal não recebe alimento ou água (Teixeira e Rodowanski, 2021). No estudo *A Multifunctional Automatic Dog-Feeder with Bluetooth and Wi-Fi Connectivity*, os autores utilizaram inúmeras tecnologias para desenvolver um sistema capaz de alimentar e fornecer água para animais de estimação. Nesse estudo, foi utilizado um microcontrolador ESP32, balança para medir o peso da ração, controle de nível de água, meios para administrar medicamentos em conjunto com a ração, sistema de refrigeração da ração e aplicativo (app) para controle do sistema. O sistema desenvolvido utiliza Bluetooth, porém consome uma corrente elevada de 34,7A, desta forma, apresenta uma grande limitação na aplicação de um sistema de nobreak pelo alto consumo de energia elétrica. Outra limitação é que não tem sistema de autenticação de animal de estimação (Boateng e Akparibo, 2022). Em outro estudo, Bemde e colaboradores (2023) utilizaram um microcontrolador Raspberry PI com sistema de câmera, de áudio, servos motores para dosar água e ração. Apesar de implementar tecnologias avançadas, o robô é imóvel e o reconhecimento por câmera é feito pelo tutor, que necessita estar conectado em tempo real para verificar se o cachorro está presente na hora de servir a ração. Outra limitação é a falta de um sistema de baterias e nobreak (Bemde et al., 2023b). Outro estudo dos mesmos autores é similar ao anterior, porém a água e a ração são controladas por temporizador. O robô também é imóvel e controla o sistema por meio de uma bateria de 12 V CC; porém, não possui sistemas de carregamento de bateria e nobreak, além de não ter um sistema de reconhecimento

de animais de estimação (Bemde et al., 2023a).

Na atual circunstância, faz-se necessário desenvolver um dispositivo capaz de identificar animais e oferecer alimento na ausência do tutor. Ademais, mesmo na presença do dispositivo dotado de inteligência artificial (IA), capaz de autenticar o animal, e automaticamente servir a ração, a falta de energia elétrica pode comprometer o funcionamento do sistema, levando a problemas de alimentação supracitados. Portanto, este estudo tem como objetivo desenvolver um dispositivo inteligente capaz de auxiliar os tutores de animais que precisam se ausentarem e têm dificuldade em deixar seu animal de estimação em casa. O dispositivo terá a capacidade de distinguir animais diferentes em um mesmo ambiente, oferecer água e alimento nas condições e quantidades pré-estabelecidas, além de controlar a falta de energia elétrica por meio de um nobreak inédito com autonomia de mais de 48 horas.

2 JUSTIFICATIVA

Considerando os cuidados diários com animais domésticos, essa dissertação apresenta o desenvolvimento de um sistema inteligente que permite ao tutor ou uma família tratar e interagir com o seu animal de estimação por intermédio do controle de ração, água e iluminação, mesmo na ausência do tutor e na falta de energia elétrica por longos períodos. O sistema trará maior segurança para o animal de estimação e o imóvel, considerando que não será mais necessário fornecer as chaves da residência a terceiros para supervisionar e alimentar os animais. A solução dessa dissertação, comparada o com produtos similares, é feita deixando em evidência as características e vantagens deste novo conceito de relação entre homem e animal de estimação, tais como:

- Tranquilidade para o tutor e segurança para o animal.
- Controle das principais necessidades dos animais domésticos, proporcionando maior comodidade nos cuidados via internet, de qualquer lugar do mundo. Total controle de água, ração e luminosidade, mesmo na ausência de energia elétrica.
- Controle e monitoramento de qualquer aparelho eletrônico: smartphones, notes, computadores, etc.
- Reconhecimento, diferenciação e autenticação do animal de estimação com aplicação de IA com captura via câmera e classificação através de rede neural.
- Nobreak com mais de 48 horas de autonomia.
- Sistema de alarme local e discadora para celular na falta de corrente alternada.

3 OBJETIVOS

3.1 OBJETIVO GERAL

O objetivo deste trabalho é criar um sistema inteligente controlado via web que contenha um hardware, software e mecânica capaz de classificar, por meio de IA, animais diferentes em um mesmo ambiente e oferecer alimento e água de forma personalizada, mesmo na ausência de energia elétrica e do dono.

3.2 OBJETIVOS ESPECÍFICOS

- Criar uma estrutura mecânica de alimentador e dosagem de água.
- Criar placas eletrônicas do *Datalogger* e do nobreak dedicado a animais de estimação.
- Desenvolvimento da IA para classificação de imagem de diferentes animais de estimação em um mesmo ambiente.
- Desenvolvimento de um nobreak inédito dedicado ao sistema de classificação, identificação e reconhecimento de animais de estimação para evitar a queda repentina de energia.

4 ORGANIZAÇÃO DA DISSERTAÇÃO

FUNDAMENTOS TEÓRICOS – Neste capítulo, é apresentado o embasamento teórico necessário para execução desse trabalho. Desta forma, dará o devido suporte aos leitores quanto ao estado atual da arte tratado nesta dissertação;

REVISÃO DE LITERATURA – Na revisão bibliográfica, são apresentados os estudos das tecnologias aplicadas nos cuidados dos animais de estimação via web;

MATERIAIS E MÉTODOS – São descritos a metodologia, materiais e aplicação das tecnologias;

RESULTADOS E DISCUSSÃO - São apresentados os resultados das medições efetuadas com o dispositivo e discussão dos resultados;

CONCLUSÃO - São apresentadas as principais conclusões do trabalho;

PERSPECTIVAS FUTURAS - São apresentadas sugestões para trabalhos futuros.

5 FUNDAMENTOS TEÓRICOS

Neste capítulo, é abordado o embasamento teórico necessário para o entendimento de conceitos importantes desta dissertação.

5.1 IOT – INTERNET DAS COISAS

Em 1999, Kevin Ashton cunhou o termo IoT ao explorar dispositivos de Identificação por Rádio Frequência (RFID) e sensores de rede sem fio. A IoT representa a interconexão digital de objetos físicos capazes de medir, coletar e transmitir dados remotamente pela internet (Ashton, 2010).

A IoT é aplicada em diversas áreas, na indústria, no comércio, bem como em residências. A automação residencial é uma área crescente de pesquisa e desenvolvimento com potencial para transformar a gestão e operação dos lares. A IoT integra dispositivos inteligentes, sensores e atuadores (Molina, 2024).

Em 2013, um avanço significativo na aplicação da tecnologia da informação na automação residencial foi alcançado com o desenvolvimento de um alimentador controlado remotamente (Jacomini, 2013). Este projeto inovador permitiu o controle de água, ração, iluminação e até mesmo a interação com animais de estimação. O alimentador foi projetado para dosar água diretamente da caixa d'água ou da rua. O processamento foi realizado por um microcomputador conectado a uma placa na porta paralela, que controla três relés. Para facilitar o acesso aos bits da porta paralela, foi desenvolvido um programa em Delphi, apresentando ícones intuitivos para água, ração e iluminação. Além disso, o dispositivo incluía uma câmera de monitoramento conectada a uma placa PCI de captura de vídeos. O acesso ao sistema era realizado remotamente através de ferramentas como VNC, *TeamViewer* ou *AnyDesk*, destacando as limitações tecnológicas da época em relação ao controle via web (Jacomini, 2013).

No estudo intitulado "Estudo dos Agronegócios 4.0", os autores, por meio de uma revisão bibliográfica, delinearam a evolução do agronegócio desde 1900 até os dias atuais, bem como os benefícios da adoção de tecnologias da Indústria 4.0 nesse setor. Desde os primeiros dias da tração animal até a era atual da agronomia, observa-se uma transformação radical impulsionada pela aplicação de tecnologias como IoT, Big Data, Sensores Wi-Fi, Drones, IA e Aprendizado de Máquina (Oliveira et al., 2022).

5.2 MQTT (MESSAGE QUEUING TELEMETRY TRANSPORT)

A *Amazon Web Services* (AWS) IoT disponibiliza um guia para desenvolvedores de soluções IoT, onde são explicados os significados das siglas MQTT em português e em inglês: Enfileiramento de Mensagens e Transporte de Telemetria (*Message Queuing Telemetry Transport*), respectivamente. Esse protocolo é essencial para a comunicação de máquina para máquina (M2M - *Machine to Machine*), com foco em IoT, e opera utilizando a estrutura do protocolo TCP/IP. Desenvolvido inicialmente pela IBM na década de 90 para preencher a necessidade de um protocolo leve capaz de permitir a comunicação entre várias máquinas, o protocolo MQTT ganhou popularidade por sua eficiência em termos de largura de banda, evitando congestionamentos de rede e permitindo uma comunicação full-duplex. A comunicação era estabelecida utilizando microcontroladores para controle e sensoriamento entre as máquinas (Amazon Web Services, 2024).

Um exemplo prático do uso do protocolo MQTT é o *Dogs Feed Smart System with Food Scales Indicator IoT Based*, um alimentador inteligente repleto de recursos, gerenciado através desse protocolo (Vimos et al., 2022). O dispositivo inclui um Arduino Uno R3 responsável pelo controle do servomotor para a dispensação da ração, um relógio de tempo real, um leitor *Radio Frequency Identification* (RFID) na coleira para identificação do cachorro e liberação de acesso à ração, além de um módulo de célula de carga para medir a quantidade correta de ração a ser dispensada. Um microcontrolador ESP8266 recebe todos os sinais do Arduino e os envia para a nuvem via protocolo MQTT. Os usuários podem acessar e configurar o alimentador por meio de um aplicativo Android que se conecta ao dispositivo usando o protocolo MQTT. No aplicativo, os usuários podem selecionar a quantidade de ração a ser dispensada e quantas vezes ao dia desejam alimentar o animal de estimação. Este sistema resolve o problema comum de falta de alimentação para animais de estimação quando o tutor está ausente, garantindo que o animal seja alimentado automaticamente nos horários configurados e autenticado pela coleira *RFID* do cachorro. Dada a frequência de múltiplos animais de estimação em uma mesma casa, cada um com suas próprias necessidades alimentares, essa abordagem é particularmente interessante, pois permite que cada animal receba sua ração individualmente (Harahap et al., 2022).

5.3 AWS IOT – AMAZON

A AWS IoT também fornece serviços de nuvem para conectar dispositivos IoT, bem como fornece software para dispositivos se integrarem às soluções AWS IoT (Amazon Web Services, 2024). Trata-se de uma plataforma de computação em nuvem oferecida pela Amazon.com. Nela, há uma variedade de serviços de computação, armazenamento, rede, banco de dados, análise, IA, *machine learning*, IoT, segurança e entre outros.

Um estudo teve como objetivo eliminar a leitura manual de consumo de energia elétrica nas residências. Assim, o dispositivo atenderia os objetivos de evitar erros humanos na leitura, de enviar alarmes de suspeita de roubo de energia elétrica em casos de anomalia na média de consumo e enviar a fatura automaticamente no celular dos clientes. Os autores usaram um microcontrolador STM32 para receber a leitura do sensor eletrônico PZEM-004T que é voltímetro, amperímetro e um wattímetro. O Sensor é ligado ao 220 V AC do relógio medido de energia e tem uma interface de saída serial nível TTL – 5 V CC. Essa saída serial RS-232 é conectada ao STM32, que calcula tensão, corrente e potência consumida, envia para a nuvem AWS (Devaki et al., 2023). Esses sistemas foram amplamente utilizados no desenvolvimento do dispositivo descrito nesta dissertação.

5.4 ML – MACHINE LEARNING (APRENDIZADO DE MÁQUINA)

A definição de "Machine Learning" (ML), ou "Aprendizado de Máquina" em português, representa um campo de estudo dentro da IA e da ciência da computação. ML utiliza algoritmos para aprender e aprimorar seu desempenho, emulando o processo de aprendizagem humano. Em 1962, Arthur Samuel criou o termo "Machine Learning" ao desenvolver um sistema computacional capaz de derrotar o melhor jogador de damas da época (Samuel, 1959).

Uma ilustração da implementação de Machine Learning foi a aplicação na classificação de raças de cachorro. Os autores empregaram técnicas da plataforma *Edge Impulse* para este fim. Aplicaram uma *Raspberry Pi* equipada com uma câmera para detectar e classificar os cachorros em ambientes residenciais. A detecção dos cachorros foi realizada utilizando a rede neural YOLOv5, enquanto a classificação das raças foi feita com a biblioteca de aprendizado de máquina ML.Net. Após a captura,

processamento e classificação das imagens, o sistema envia uma mensagem de texto (SMS) aos usuários com os resultados obtidos (Macedo, 2023). Similarmente, o presente estudo implementou algoritmos de IA para treinar redes neurais de classificação e identificação de diferentes animais de estimação em um mesmo ambiente.

5.5 REDE NEURAL

A concepção do neurônio artificial, derivado do neurônio biológico, foi apresentada por Warren McCulloch e Walter Pitts em 1943, marcando a introdução do primeiro modelo "computacional" de neurônio. Essa inovação abriu caminho para a pesquisa em redes neurais, que se dividiu em duas abordagens distintas: uma voltada para os processos biológicos no cérebro e outra direcionada às redes neurais na área da IA (McCulloch e Pitts, 1943).

Redes neurais artificiais (RNAs) constituem um conjunto de algoritmos de IA que se baseiam em um modelo inspirado no neurônio humano, sendo capazes de aprender por meio de experiências e estímulos. O aprendizado de máquina em sistemas embarcados, conhecido como *tinyML*, adapta os princípios do aprendizado de máquina para que sejam viáveis em dispositivos de borda, ou seja, dispositivos com capacidade reduzida de memória e processamento. O *tinyML* foi desenvolvido para atender à necessidade de processamento de dados em tempo real, com baixo custo e sem a necessidade de enviar dados para a nuvem. Além disso, com a integração da IA à IoT, surgiu uma nova geração de dispositivos inteligentes, conhecida como *Artificial Intelligence of Things* (AIoT), que combina a coleta de dados por meio de dispositivos IoT com a capacidade de tomar decisões localmente em tempo real, utilizando o aprendizado de máquina *tinyML* (Nascimento, 2022)..

5.6 EDGE IMPULSE

Ainda sendo uma ideia nova, o *Edge Impulse* também é uma plataforma de treinamento de redes neurais projetada para *TinyML*. O *Edge Impulse* foi fundado em 2019 por Zach Shelby e Jan Jongboom com a missão de pavimentar o caminho para o futuro do aprendizado de máquina incorporado, capacitando os desenvolvedores a

criar e otimizar soluções com dados do mundo real. Os criadores têm como objetivo simplificar o processo de construção, implantação e escalonamento de aplicativos de ML incorporados, tornando-o mais fácil e rápido do que nunca. Além disso, a plataforma foi reconhecida com os prêmios de Melhor em plataformas de aprendizado de máquina em 2021, 2022 e o Prêmio AI Tech em 2023. Esses reconhecimentos destacam sua inovação e impacto na engenharia e ciência de dados. Portanto, quando se trata de IA para embarcados atualmente, o *Edge Impulse* certamente é uma referência. Ao longo dos últimos anos, tem contribuído e sido aplicado em diversas áreas da ciência, podendo ser implantado em uma ampla variedade de hardware. Por meio do link <https://edgeimpulse.com/>, os usuários podem criar uma conta e utilizar a plataforma para treinar seu próprio hardware de borda. A plataforma oferece um conceito de prototipagem de software rápida, permitindo que, após o treinamento do modelo, seja escolhido em qual microcontrolador o algoritmo será executado (Hymel et al., 2023).

Uma aplicação prática dessa tecnologia é o robô doutor humanoide recentemente apresentado. Utilizando tecnologias avançadas, como IoT, chatGPT e *Edge Impulse*, o autor afirma que a plataforma *Edge Impulse* é um sistema de aprendizagem baseado em IA capaz de integrar dispositivos de borda para coletar sintomas do paciente, como áudio, imagem e texto. No contexto deste trabalho, o *Edge Impulse* foi aplicado para classificar duas doenças com base em sinais de áudio: pneumonia e asma. Após o treinamento com amostras de áudio dessas doenças, foi possível desenvolver um modelo classificador eficaz para distinguir entre elas (Venkataswamy, Janamala e Cherukuri, 2023).

5.7 MECÂNICA – DIFERENTES MECANISMOS DE DOSAGEM DE RAÇÃO

A análise do estado da arte revela a presença de dosador de ração por gravidade, onde o animal tem acesso livre à comida (FERCAR, 2024).

Um outro tipo de dosador foi proposto pelos autores do trabalho denominado Inovação na alimentação de animais de estimação: Desenvolvimento de um sistema automatizado para animais de estimação. Nele foi integrado um sistema de armazenamento de ração e um mecanismo de transporte. Esse mecanismo consiste em um tubo equipado com uma engrenagem de rosca sem fim, acionado por um motoredutor de corrente contínua, responsável por transferir a ração do reservatório

para o recipiente de distribuição (Pereira et al., 2023).

Diferentemente, no trabalho chamado *Feeding Pets using Automation Technology and Google Assistant*, os autores utilizam servidores *NodeMCU* e *Network Time Protocol* (NTP) para responder aos comandos do comedouro enviados via *Google Assistant*, fornecendo alimentação por meio de controle por servo-motor (Prasad et al., 2023).

Por outro lado, os autores do projeto “Alimentador Automático de Animais Domésticos Utilizando” implementaram um controle de Motor de Passo dotado de sensor encoder angular para controlar o posicionamento do eixo do motor através dos passos nas bobinas A, B, C e D (Pires et al., 2022).

O mecanismo adotado nesta dissertação difere dos outros existentes no estado da arte, apresentando uma abordagem inovadora para a dosagem de ração. Utilizando um tubo de PVC cortado ao meio para armazenar a quantidade desejada de ração, a dosagem é realizada de forma giratória, acionada pelo eixo de um motoredutor. Uma única rotação completa de 360° dispensa a quantidade necessária no recipiente de comida do cachorro.

Consequentemente, o uso de ferramentas de IoT, redes neurais e mecanismos de dosagem permite a construção de um sistema integrado que visa fornecer alimento e água para animais de estimação na ausência de tutores.

6 REVISÃO DA LITERATURA

Nesta seção, são apresentados estudos que investigaram a aplicação de tecnologias na área de cuidados, controle e monitoramento de animais de estimação.

6.1 MERCADO PET NO BRASIL

Existem inúmeros fatores responsáveis pelo crescimento do mercado de animais de estimação no Brasil. O maior deles consiste no aumento da quantidade desses animais nos lares brasileiros, que tem impulsionado o crescimento deste setor, tornando-o o terceiro maior do mundo em faturamento. Além disso, tutores priorizam a saúde, o bem-estar e a alimentação de seus animais, destacando a importância da qualidade dos produtos e serviços (Dalmas, 2019). Da mesma forma, a expansão do mercado de animais de estimação no Brasil está intimamente ligada às transformações socioeconômicas da população e aos avanços na medicina veterinária observados nos últimos anos. Como essas mudanças impactam o crescimento nesse mercado, destaca-se a importância do marketing para a análise de mercado e a construção de estratégias eficazes para alcançar esse público consumidor (Elizeire, 2013). Benedito (2018) afirma que esse crescimento tem sido impulsionado pela maior presença dos animais de estimação nos lares e pelos vínculos afetivos mais fortes entre humanos e seus animais de estimação (Benedito, 2018).

6.2 UTILIZAÇÃO DE SISTEMAS INTELIGENTES PARA CUIDADOS DE PETS

Quanto à aplicação de tecnologia nos cuidados com animais de estimação, existem inúmeros sistemas descritos. Por exemplo, ao aplicar um controle com sistema embarcado simples e processamento local, Jain e colaboradores (2023) utilizaram um Arduino Pro Mini no controle de identificação por rádio frequência (RFID), relógio de tempo real (RTC) e servomotor. O projeto controla apenas a ração, com o objetivo de ensinar o cachorro a se alimentar no horário correto e controlar a quantidade para prevenir obesidade (Jain et al., 2023).

Similarmente, Rossi, Joselino e Junior (2017) desenvolveram um alimentador

automático de ração para cachorro, incluindo estrutura mecânica, eletrônica e programação do Arduino. O dosador de ração foi controlado por motor de passo e uma rosca sem fim que transporta a ração do reservatório até a bandeja de alimentação. A parte eletrônica foi controlada por Arduino e IHM, no qual o usuário define quantidade e horários de dispensação da ração. O sistema também conta com um reservatório local de água que dosa água através de uma válvula solenoide 110 V AC (Rossi, Joselino e Junior, 2017).

Quando se trata de identificação e diferenciação de animais, Teixeira e Rodowanski (2021) criaram um alimentador para cães de rua com identificação por RFID, onde uma tag na coleira do animal permite a autenticação quando se aproxima do alimentador. A tampa controlada por Arduino e um motoredutor de corrente contínua abre até o ponto onde a ração está localizada, permanecendo aberta enquanto o cão se alimenta (Teixeira e Rodowanski, 2021). Sistemas como esse também são relevantes no contexto da alimentação de animais de estimação em casa.

Já Boateng e Akparibo (2022) apresentaram uma solução com controle de temperatura e refrigeração do local de armazenamento da ração. O projeto integra dispensação de água, ração e medicamentos secretamente junto com a ração, utilizando um microcontrolador ESP32 para conexão Wi-Fi, Bluetooth, controle IoT, e um programa para celular desenvolvido na *IDE App Inventor* (Boateng e Akparibo, 2022). Esta solução foi interessante, pois além dos cuidados com a água e a ração, incorporou a possibilidade de administrar medicamentos à distância, garantindo maior tranquilidade a tutores.

Seguindo pela mesma linha de solução, Harahap e colaboradores (2022) aplicaram IoT para o controle de água potável e ração, bem como adicionaram uma célula de carga no recipiente de ração para dosar a quantidade ideal no horário de alimentação do animal. Além disso, sensores ultrassônicos dosaram a quantidade necessária de água potável. O sistema foi controlado por um Arduino Uno R3 e uma ESP8266 para a conexão Wi-Fi (Harahap et al., 2022). Com solução parecida com as anteriores, porém com uma abordagem diferente, Bemde e colaboradores (2023) desenvolveram um robô creche para animais de estimação utilizando sensores e *Raspberry Pi*, com capacidade para controlar água e ração do cachorro, enviando informações para uma aplicação que gerencia a quantidade dispensada em intervalos determinados (Bemde et al., 2023b).

No entanto, Faria e colaboradores (2023) implementaram um sistema distinto

dos demais. Os autores implementaram sistemas com IA, demonstrando a viabilidade de um modelo de reconhecimento de aves usando a plataforma *Edge Impulse*, alcançando a precisão esperada de classificação e auxiliando na alimentação dos patos. Os autores aplicaram uma ESP32 com câmera para atender o projeto com custo significativamente baixo, tornando a tecnologia acessível para aplicação na avicultura (Faria, Melo e Muller, 2023).

A IoT também tem sido uma solução essencial no controle e monitoramento de ração e água. Um estudo recente utilizou IoT para monitorar e gerenciar um comedouro para animais de estimação conectado à internet. O sistema permitia alimentação remota através de uma página web, usando microcontroladores Arduino Uno e ESP8266, sensores ultrassônicos, motor de passo, relógio em tempo real e sensor de peso (BIRHA *et al.*, 2022). Outro estudo apresentou o uso de um Dispenser, um dispositivo inteligente que simplifica a rotina de cuidados com animais de estimação, proporcionando alimentação automática. Composto por estrutura mecânica, sistema embarcado com Arduino Mega e servidor de aplicações com *Raspberry Pi*, o dispositivo permite monitoramento e controle remoto via aplicativo conectado à rede (Rodrigues *et al.*, 2024). No quesito integração e controle de dados, um estudo desenvolveu um protótipo de comedouro inteligente utilizando IoT, ESP32, sensor ultrassônico e servomotor para dosagem de ração controlada por um aplicativo móvel, com comunicação em tempo real através de um banco de dados (Ramos e Junior, 2023).

No contexto de animais de rua, Silva (2021), com objetivo de resolver o problema de cachorros e gatos abandonados na pandemia, desenvolveu um alimentador colaborativo para cães e gatos de rua usando IoT e plataforma NodeMCU para envio de dados via Wi-Fi. O sistema incluiu sensores de célula de carga e nível de água, com um aplicativo Android para visualização e reabastecimento comunitário (Silva, 2021).

Ainda como solução aplicando a IoT, Rodrigues e colaboradores (2019) construíram um alimentador automático para animais de estimação que mede e exibe a quantidade de ração, controlado por um servidor broker MQTT conectado à rede Wi-Fi (Rodrigues e Queiróz, 2019). Similar ao estudo anterior, a IoT foi aplicada para monitoramento remoto do nível de água em recipientes de animais domésticos utilizando ESP32 com Shield GSM para enviar notificações de nível crítico de água para dispositivos móveis (Gadelha, 2020). Por outro lado, Chen & Elshakankiri (2020)

aplicaram IoT com servidor broker MQTT - blynk.io para implementar um sistema integrado de alimentador, dispensador de água e caixa de areia para gatos. Usando placas Arduino Uno e módulos Wi-Fi, os subsistemas são conectados à rede local, com dados exibidos em um aplicativo de smartphone para monitoramento do consumo de alimentos, água e frequência da defecação dos animais de estimação (Chen e Elshakankiri, 2020).

Os estudos citados acima mostram a diversidade de usos de sistemas mecânicos integrados ou não para a criação de alimentadores para animais de estimação. No entanto, na presente literatura, não foram identificados estudos que utilizam ferramentas integradas, redes neurais para identificação e diferenciação de animais, bem como o desenvolvimento de um nobreak dedicado a animais de estimação, que mantém o sistema em funcionamento mesmo na ausência de energia elétrica.

Tabela 1 — Comparação da dissertação com estado da arte

Autor e Ano	REURSOS									
	Água	Ração	Luz	IoT	IA	Microc.	Motor	Sensores	Datalogger	Nobreak
PIRES et al., 2022		x				ESP32	Motor de Passo	Encoder		
PIRES et al., 2022		x				Arduíno Mini	ServoMotor	RFID, RTC		
ROSSI et al., 2017	x	x				Arduíno Uno	Motor de Passo	RFID, RTC		
TEIXEIRA et al., 2021		x				Arduíno Mega	Motoredutor CC	RFID, RTC		
BOATENG et al., 2022	x	x		x		ESP32	Motoredutor CC	RFID, RTC, Peso, Peltier		
HARAHAP et al., 2022		x		x		Arduíno Uno, ESP8266	Motoredutor CC	RFID, RTC, Peso		
BEMBDE et al., 2023	x	x		x		Raspberry PI	ServoMotor	Sensor Ultrassônico		
FARIA et al., 2023		x			x	ESP32	Não Informado	Presença, Câmera		
BIRHA et al., 2022		x		x		ESP8266	Motor de Passo	Ultrassônico, RTC		
RODRIGUES et al., 2024		x		x		Arduíno Mega, Raspberry	Motor de Passo	Ultrassônico, RTC, Peso		
PIFFER et al., 2023		x		x		ESP32	ServoMotor	Ultrassônico		
SILVA, 2021	x	x		x		ESP32	Não Tem	Ultrassônico		
RODRIGUES et al., 2019	x			x		ESP32 + GSM	Não Tem	Boia, GSM		
CHEN et al., 2020	x	x		x		Arduíno Uno, ESP8266	Não Tem	Ultrassônico, Peso, Presença		
Dissertação	x	x	x	x	x	Atmega328P, ESP32, ESP32CAM	Motoredutor CC	Câmera, RTC, Peso, Boia	x	x

Fonte: Elaboração própria, 2024.

7 MATERIAIS E MÉTODOS

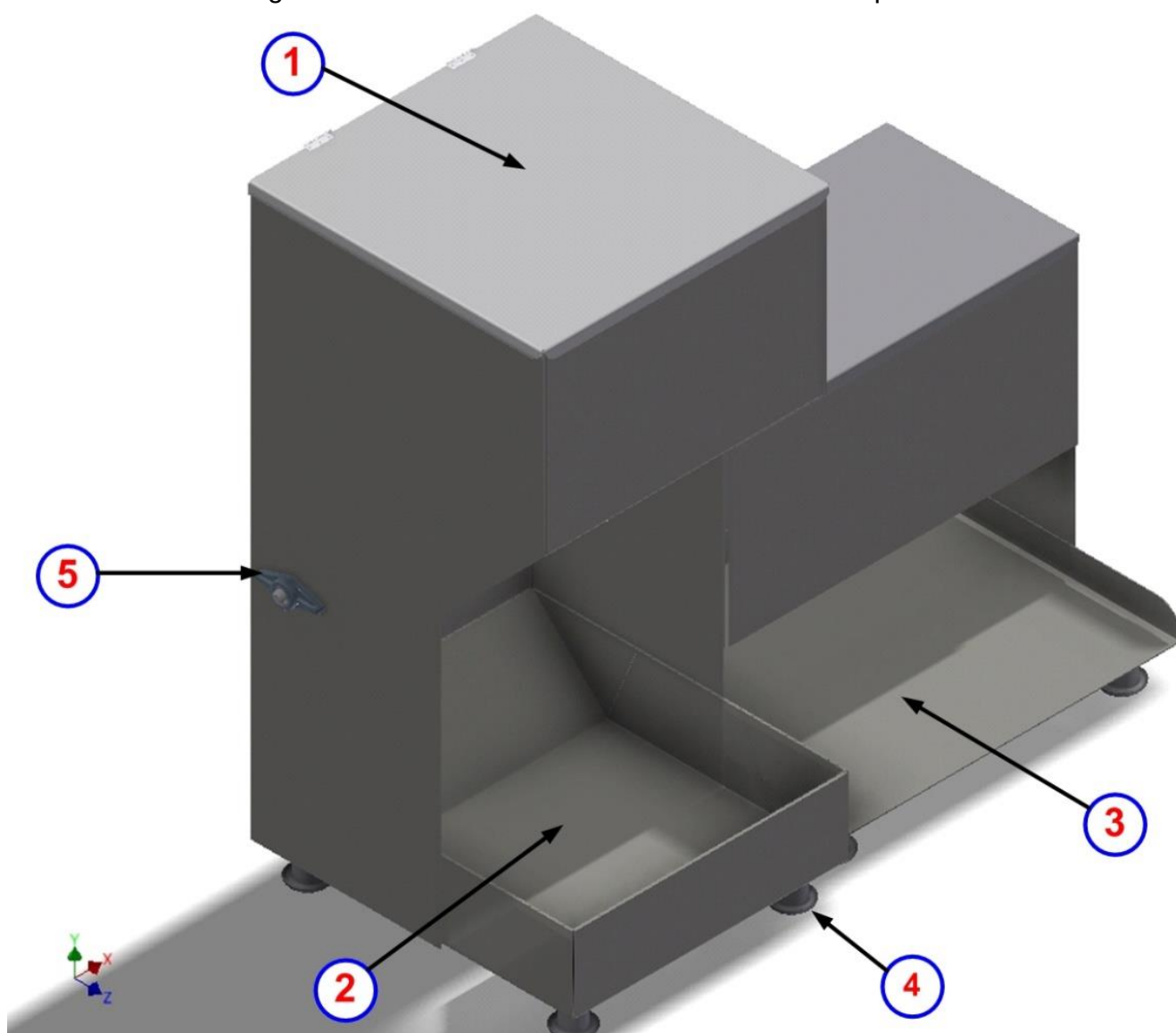
Neste capítulo, são apresentados os materiais e metodologia adotados nesta dissertação. Os métodos foram aplicados para construir um dispositivo inteligente capaz de auxiliar nos cuidados com os animais domésticos, mesmo quando o tutor não está presente. Para isso, aqui será descrito a eletrônica, a elétrica, a mecânica, a lógica de programação e o treinamento de rede neural para classificação e reconhecimento do cachorro para liberar a ração, água e mensurar suas proporções ao longo do tempo, assim como seu funcionamento na ausência de energia elétrica.

7.1 ESTRUTURA MECÂNICA DO SISTEMA

A mecânica do projeto envolve a estrutura do alimentador de ração e água. Utilizou-se o software “Solid Works” para desenvolver as peças do alimentador com a finalidade de permitir que o sistema fosse automatizado (Figura 1). Para a ração (armazenagem e dispensa), adotou-se um comedouro (Figura 2), que é vendido no mercado (Fercar). Trata-se de um comedouro por gravidade de 10 kg. Esse comedouro foi utilizado para fazer a automação, na qual foram instalados o motorreductor, fim-de-curso, mancais e o tubo cortado ao meio (Figura 3). O tubo serve para captar a quantidade de 100 gramas de ração, onde a ração é dispensada quando o motorreductor girar 1 volta completa. Ao lado direito, parte 3 do alimentador, foi construído com chapa de alumínio de 2mm de espessura. Com o desenho mecânico e dimensões dessa estrutura, foi construída a estrutura da água e local dos componentes elétricos.

A seguir, é descrito o processo de fabricação mecânica do comedor: A) medição e traçagem das partes a serem cortadas e dobradas; B) corte na máquina de serra de fita; C) retirado as rebarbas com processo de limagem; D) dobras 90° na máquina dobradeira; E) furos de 5,5mm de diâmetro; F) parafusado os dois lados da estrutura com parafusos de inox m5 com porca parloc de inox M5 e arruela m5 de inox; G) realizados furos de fixação do motor, fixação da fonte e fixação da válvula solenoide da água.

Figura 1 — Estrutura mecânica dividida em cinco partes



Fonte: Elaboração própria, 2024.

Cada estrutura mecânica possui uma função distinta dentro do sistema: 1) Local para armazenar até 10 quilos de ração. 2) Local que o alimentador dispensa ração após a dosagem via software. 3) Local que o alimentador libera água potável após a dosagem via software. 4) Pés anti formiga. 5) Mancal de apoio do eixo do motor.

Figura 4 — Detalhes da estrutura



Fonte: Elaboração própria, 2024.

7.2 ESQUEMA ELÉTRICO DO SISTEMA

Após definidos os requisitos da automação, foi idealizado o esquema elétrico (Figura 5), respeitando as devidas características elétricas do motor, lâmpada e válvula solenoide. Essa é a estrutura de potência do projeto e trabalhará na tensão de 127 V AC. Para construção do esquema apresentado na Figura 5, foi utilizado o software EasyEda online. Nele, consta o controle do motorreductor 12 V CC – 3 A, que é regulado pela fonte através de um dimmer, um transformador bivolt de 5 A e uma ponte retificadora de 10 A. No primário, a tensão é controlada em conjunto com o *dimmer*. Desta forma, o primário pode variar de 0 a 127 V AC, assim, na saída do secundário do transformador, a tensão pode ser alterada entre 0 à 12 V AC. Por fim, no secundário, tem-se uma ponte retificadora que transforma o valor de tensão de 12 V AC alternada em 12 V CC em corrente contínua. Essa fonte variável foi desenvolvida como uma forma de fazer o processamento de potência de corrente contínua, ao qual

é possível controlar a velocidade do motor.

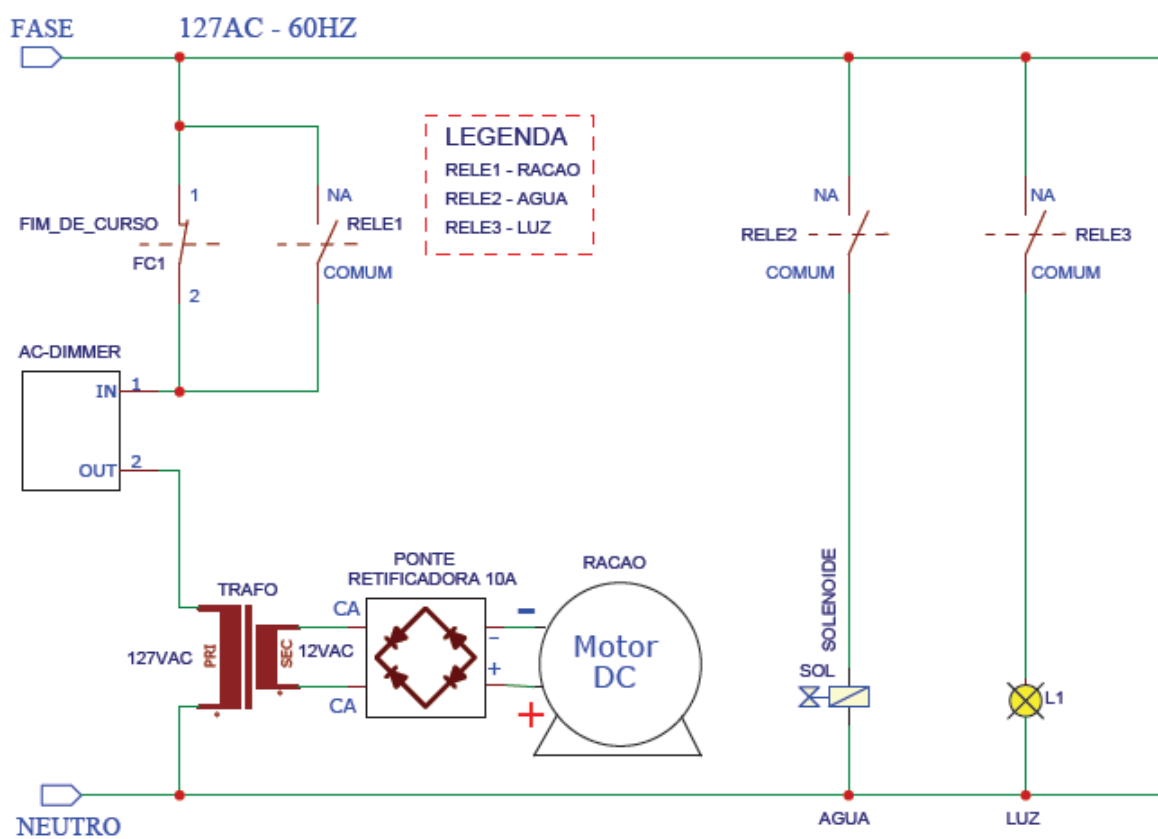
No fim-de-curso FC1, há uma função crucial. Ele foi construído de forma inteligente para poder auto-desligar o motor do alimentador de ração. Ele é um contato normal fechado e é acionado por um came colocado na posição do alimentador em modo ração contida dentro do silo. Desse modo, com o contato estando pressionado, ele mantém seu contato aberto, impedindo a passagem da fase para a fonte do motor. Em paralelo com FC1, há o contato aberto do Relé 1, que deve ser acionado pelo microcontrolador por apenas 0,5 segundos. Assim que é acionado o relé 1, ele manda a fase para o motor da ração, que começa a girar, dispensando a ração que cabe no tubo, que é 120 gramas. Nesse momento, o FC1 é desacionado e seu contato é fechado mantendo a fase na fonte do motor. Em paralelo, o relé 1 é desacionado após 0,5 segundos e o motor se desligará no momento que o FC1 for acionado novamente. Esse processamento foi idealizado considerando ser possível realizar um controle através da rede IoT; portanto, para não haver inúmeras dosagens, o microcontrolador envia um comando de ligar o relé 1 por 0,5 ms. Para mais segurança, ao pressionar o botão de ração, ele não fará nada. Se for mantido pressionado, também não fará nada, e só ligará o relé 1 após soltar o botão de ração.

O relé 2, ao ser acionado pelo microcontrolador, manda a fase à válvula solenoide, liberando a água nova e potável no pote de água do animal de estimação. Essa válvula tem uma entrada de 3/4" com rosca, que é instalada na água da caixa ou da rua. O tutor tem a opção de incluir um filtro na entrada da água. Na estrutura hidráulica, há uma válvula reguladora de vazão por haver, em muitos casos, pressão no fornecimento de água.

A lâmpada é para iluminar o ambiente do animal de estimação. Ela é de led e alimentada em 127 V AC. Ao ser acionado, o relé 3 pelo microcontrolador, envia a fase para a lâmpada acendendo a luz. Para desligar, basta pressionar o mesmo botão.

O motor possui um redutor em seu eixo, que apresenta torque suficiente para girar o dosador. Os detalhes dos elementos elétricos já posicionados e fixados no alimentador estão apresentados na Figura 6. É possível verificar o motorredutor, a solenoide, a fonte de 12 V CC do motor e a chave fim-de-curso.

Figura 5 — Esquema elétrico do alimentado



Fonte: Elaboração própria, 2024.

Figura 6 — Detalhes da montagem dos componentes elétricos



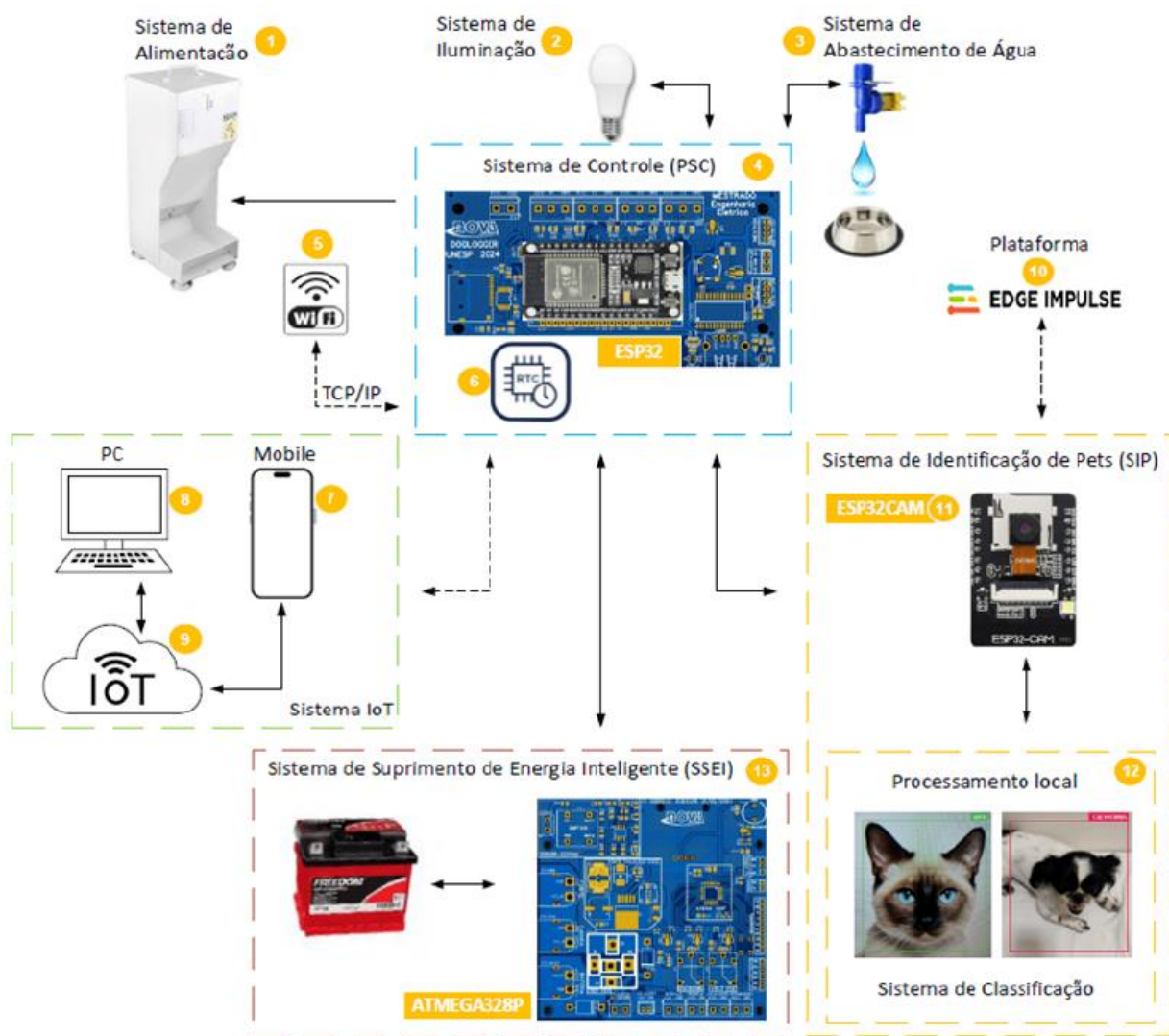
Fonte: Elaboração própria, 2024

7.3 ARQUITETURA DO SISTEMA

O diagrama de blocos da Figura 7 apresenta a arquitetura do sistema proposto, constituído por: Placa do Sistema de Controle (PSC) (4), Placa do Nobreak - Sistema de Suprimento de Energia Inteligentes (SSEI) (13) Sistema de Identificação de Pets (SIP) (11). A placa PSC gerencia os demais módulos (1 a 9) sendo responsável pela integração e processamento dos dados recebidos. PSC recebe do SSEI os valores de peso da ração, a tensão da rede AC e tensão da bateria DC. A placa SIP faz a

captação e classifica a imagem do animal de estimação por meio de técnicas de aprendizado de máquina da IA. Os dados relacionados a identificação são enviados à PSC que autoriza ou não uma dosagem de ração. Os sistemas SSEI e SIP tem seus próprios procesadores e se comunicam com a PSC através de comunicação serial. A Figura 8 apresenta a integração da comunicação entre as placas que compõem toda a arquitetura do sistema.

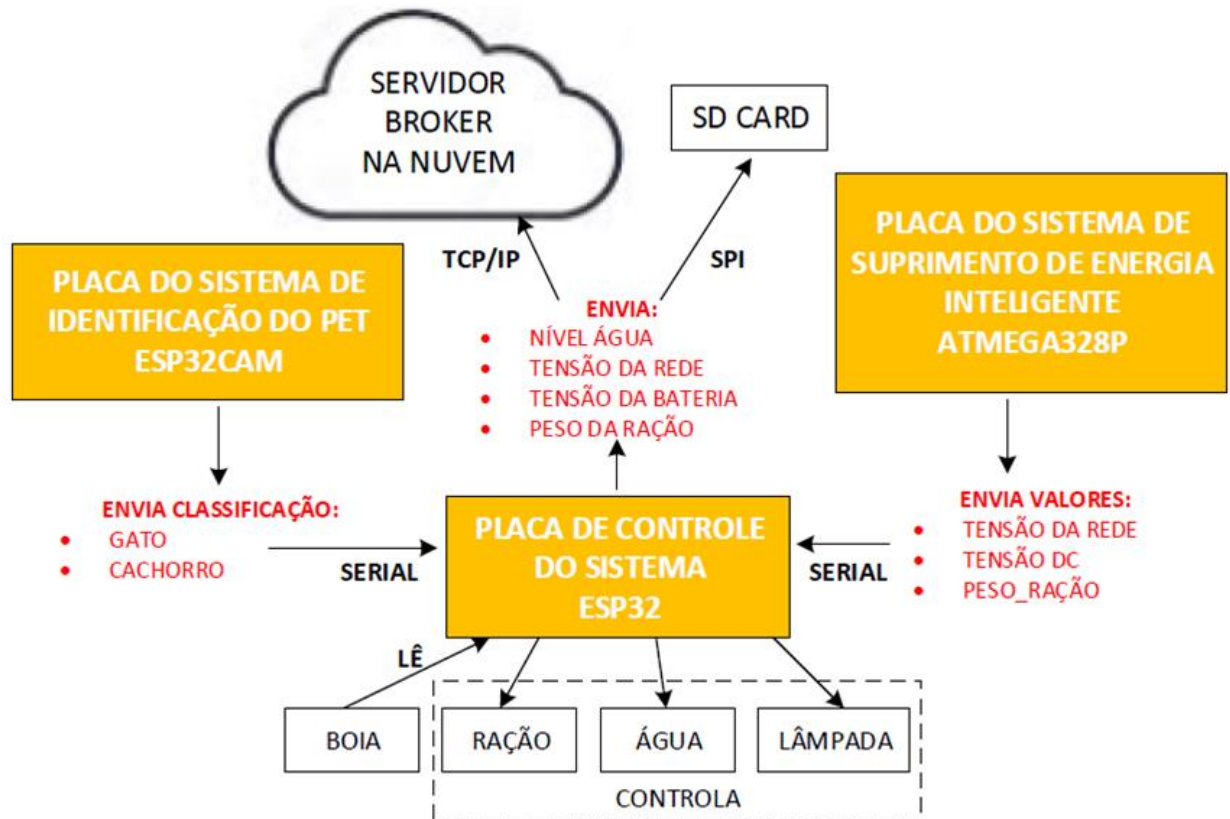
Figura 7 — Arquitetura do sistema



Fonte: Elaboração própria, 2024.

- 1) Sistema de alimentação 2) Sistema de iluminação 3) Sistema de abastecimento de água 4) Sistema de Controle 5) Wifi - TCP/IP 6) RTC 7) *Dashboard* Mobile 8) *Dashboard* PC 9) Servidor *Ubidots* 10) Plataforma *Edge Impulse* 11) Sistema de Identificação de Pets 12) Sistema de classificação e 13) Sistema de suprimento de energia inteligente.

Figura 8 — Integração da comunicação entre as placas que compõem toda arquitetura do sistema



Fonte: Elaboração própria, 2024.

7.4 PROJETO NOBREAK INTELIGENTE

7.4.1 Escopo do Nobreak Inteligente

O controlador do nobreak adota o microcontrolador Atmega328P (Microchip), conforme descrito na folha de dados do componente (Atmel Corporation, 2015). Este microcontrolador é de 8 bits e segue a arquitetura Harvard, fornecendo 23 portas de entrada/saída, memória flash de 32KB, 2KB de memória SRAM e 1KB de EEPROM. Ele é o mesmo processador usado no Arduino Uno e Nano, sendo utilizado aqui apenas o chip com cristal, capacitores e resistores, todos onboard.

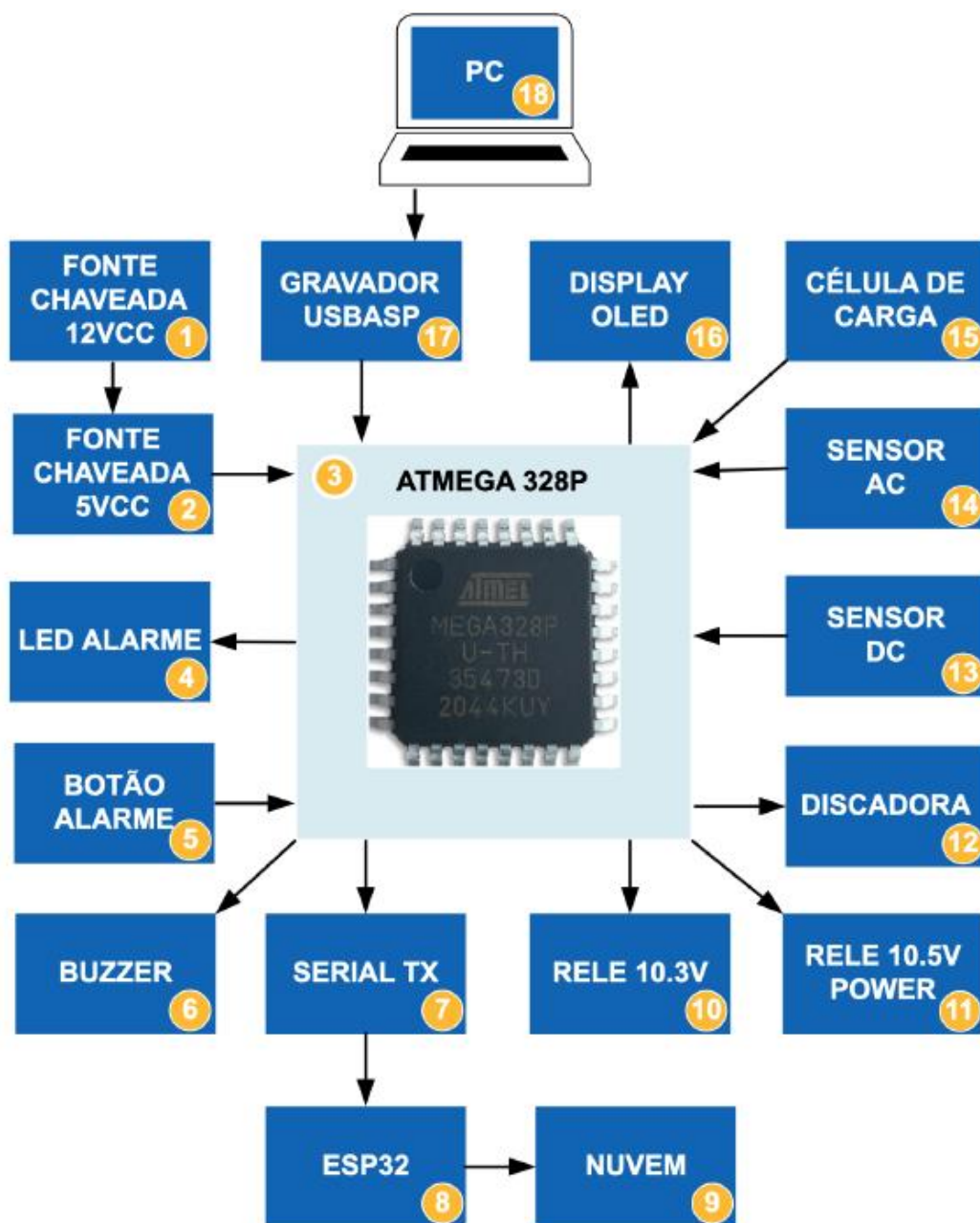
O sistema do nobreak realiza a leitura da bateria/fonte, do valor de AC e mede o peso da balança. Controla os níveis mínimo e máximo de tensão da bateria e possui dois tipos de alarmes: 1) Falta de AC. 2) Problema na fonte. Além disso, conta com

uma discadora para celular e envia na serial TX o valor calculado de AC, DC e Peso da Ração.

Para o controle de baixa tensão na bateria em caso de falta de AC, o nobreak opera em dois níveis: 1) Mantém a potência ligada enquanto a bateria estiver com tensão superior a 10.5 V CC. Caso a tensão caia abaixo desse valor, é desacoplada a potência do circuito da bateria por meio de um relé de 60 A. 2) Os controladores permanecem ligados enquanto a bateria estiver com tensão superior a 10.3 V CC. Se a tensão da bateria cair abaixo de 10.3 V CC, outro relé é desacionado, desligando todo o circuito. O sistema só é reativado quando o 127 V AC retornar, acionando a fonte de 14.4 V CC, os controladores e a parte de potência, além de recarregar a bateria.

A placa foi elaborada na plataforma EasyEda, com a vasta maioria dos componentes em montagem superficial (SMD), e a programação foi realizada no ambiente de desenvolvimento integrado (IDE) do Arduino. Na Figura 9, é possível visualizar o resultado da placa de circuito impresso (PCI), enquanto a Figura 10 apresenta todos os detalhes para a ligação do nobreak. Além disso, na Figura 8 a seguir, é apresentado o diagrama de blocos que representa de maneira abrangente o escopo do Nobreak Inteligente, oferecendo uma visão geral dos diversos recursos que ele possui.

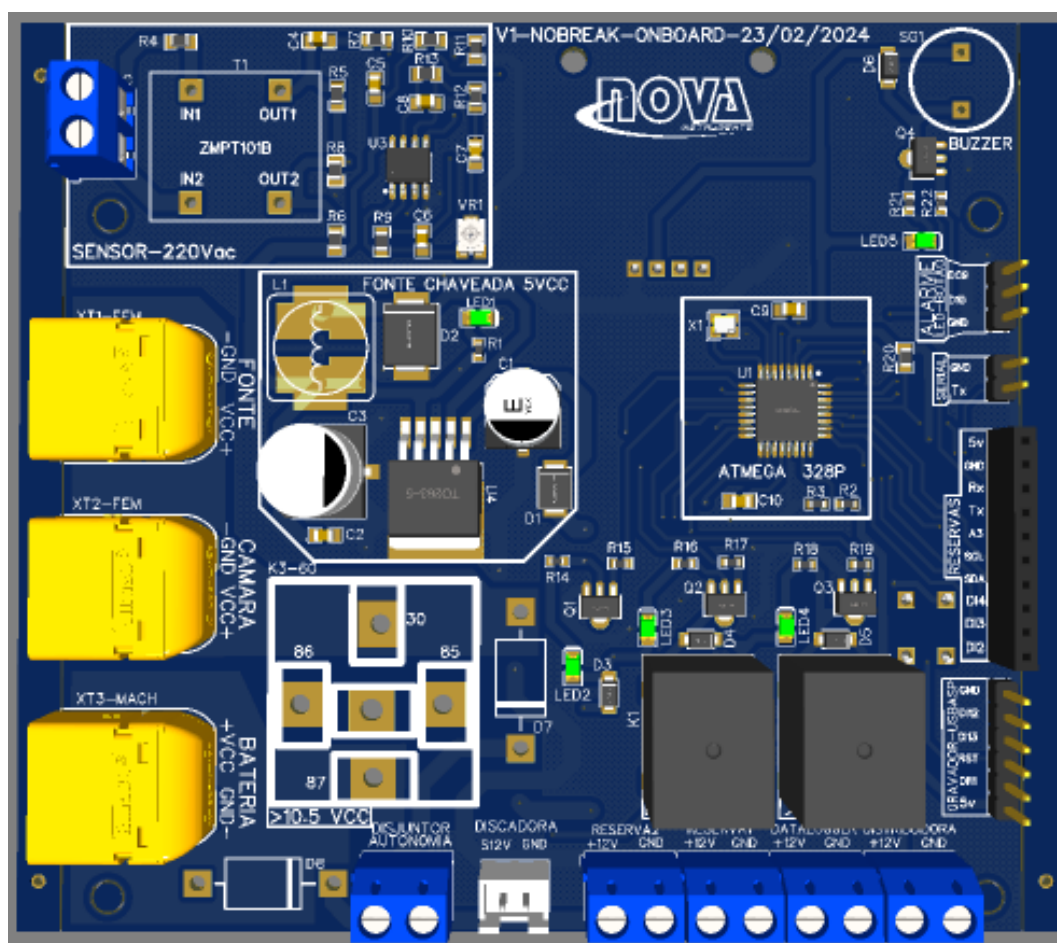
Figura 9 — Diagrama em bloco do Hardware do Nobreak Inteligente



Fonte: Elaboração própria, 2024.

Após a definição do escopo do projeto do nobreak, o esquema eletrônico foi elaborado utilizando o software EasyEda. Em seguida, o projeto do circuito impresso (PCB) foi desenvolvido, os componentes SMD foram selecionados e o pedido de produção foi encaminhado para a empresa JLCPCB, localizada na China. Na Figura 10 abaixo, é possível observar a placa em uma representação tridimensional.

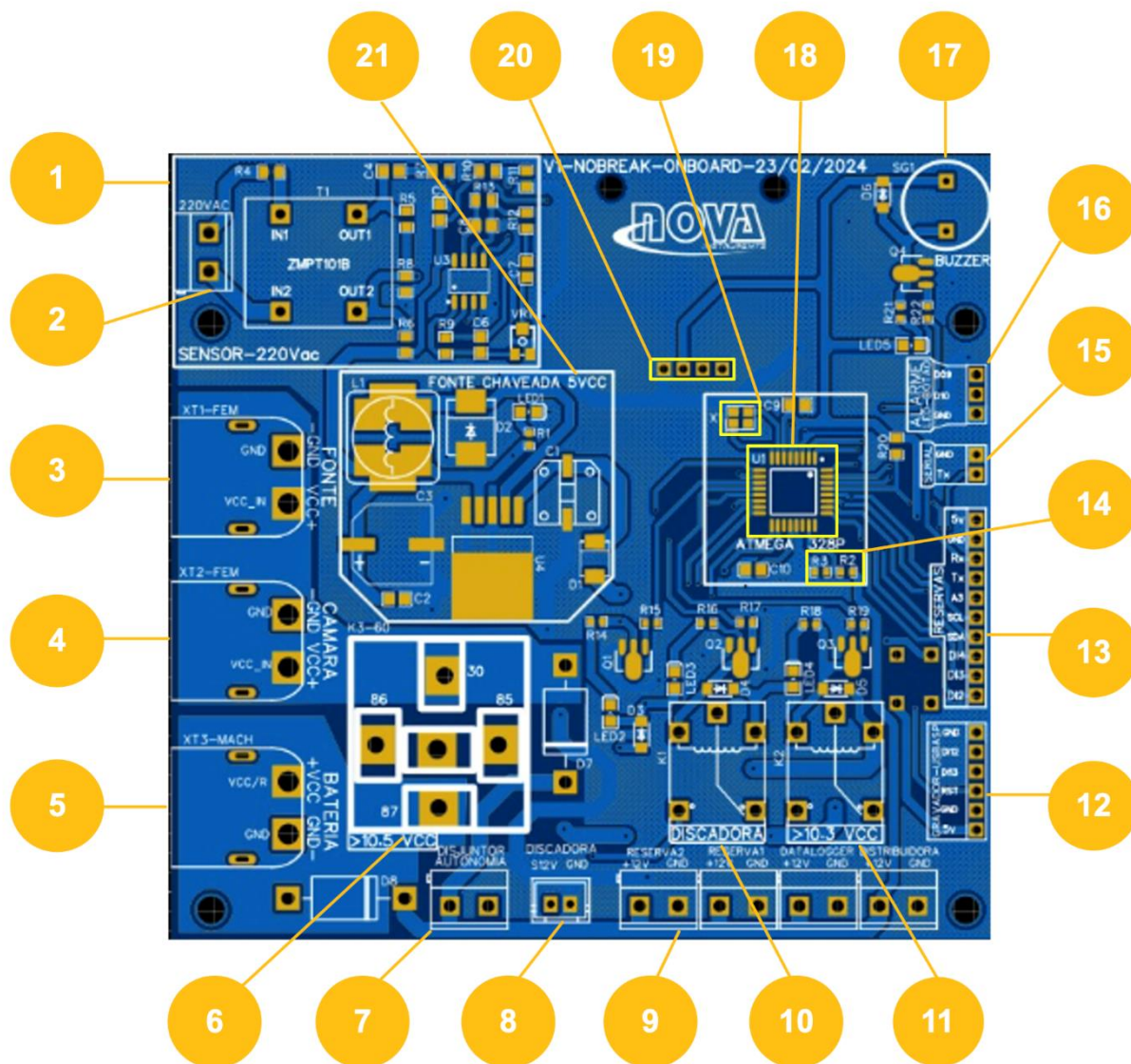
Figura 10 — vista em 3D da PCI do Nobreak no Software Easyeda



Fonte: Elaboração própria, 2024.

A próxima Figura 11 em vista 2D contém todos os detalhes da PCB do Nobreak Inteligente e seus devidos bornes de ligação do circuito elétrico.

Figura 11 — Vista em 2D do nobrek com todos os detalhes



Fonte: Elaboração própria, 2024.

- 1) Parte do sensor AC onboard.
- 2) Conector onde é a entrada AC de 0 à 220 V AC.
- 3) Conector da entrada da fonte que alimenta o circuito e carrega a bateria.
- 4) Conector que liga a parte de potência do circuito (Motor da Ração).
- 5) Conector da Bateria.
- 6) Relé de 60 A – Liga e desliga a bateria da carga.
- 7) Liga e desliga do Nobreak.
- 8) Sinal de saída da discadora.
- 9) Quatro tomadas de 12 V CC.
- 10) Relé da discadora – Alarme falta de AC.
- 11) Relé que controla e protege a bateria – Corta com valor mínimo da autonomia da bateria.
- 12) Seis terminais de Gravação In-Circuit (Gavador USBASP).
- 13) Pinos reserva do Atmega 328P.
- 14) Sensor Dc - Rede de 2 resistores em série.
- 15) Transmissão Serial – USART TX – Envia os dados para o Datalogger.
- 16) GND + Led + Botão silenciar do alarme.
- 17) Buzzer sonoro do alarme.
- 18) Microcontrolador Atmega 328P.
- 19) Cristal de 16 MHZ.
- 20) Quatro pinos de conexão do display Oled.
- 21) Fonte chaveada – conversor CC-CC do tipo buck de 12 V CC para 5 V CC de 3 A.

Abaixo, são descritas informações sobre a utilização do gravador USBASP e os detalhes do projeto de todos os sensores do Nobreak inteligente. Os componentes abordados são os seguintes:

- A. Gravador USBASP.
- B. Sensor AC Voltage LM358
- C. Sensor de tensão DC da fonte/bateria.
- D. Sensor de célula de carga - balança da ração.

Essas informações serão apresentadas de maneira detalhada para proporcionar uma compreensão abrangente do funcionamento do sistema.

7.4.2 Gravador USBASP

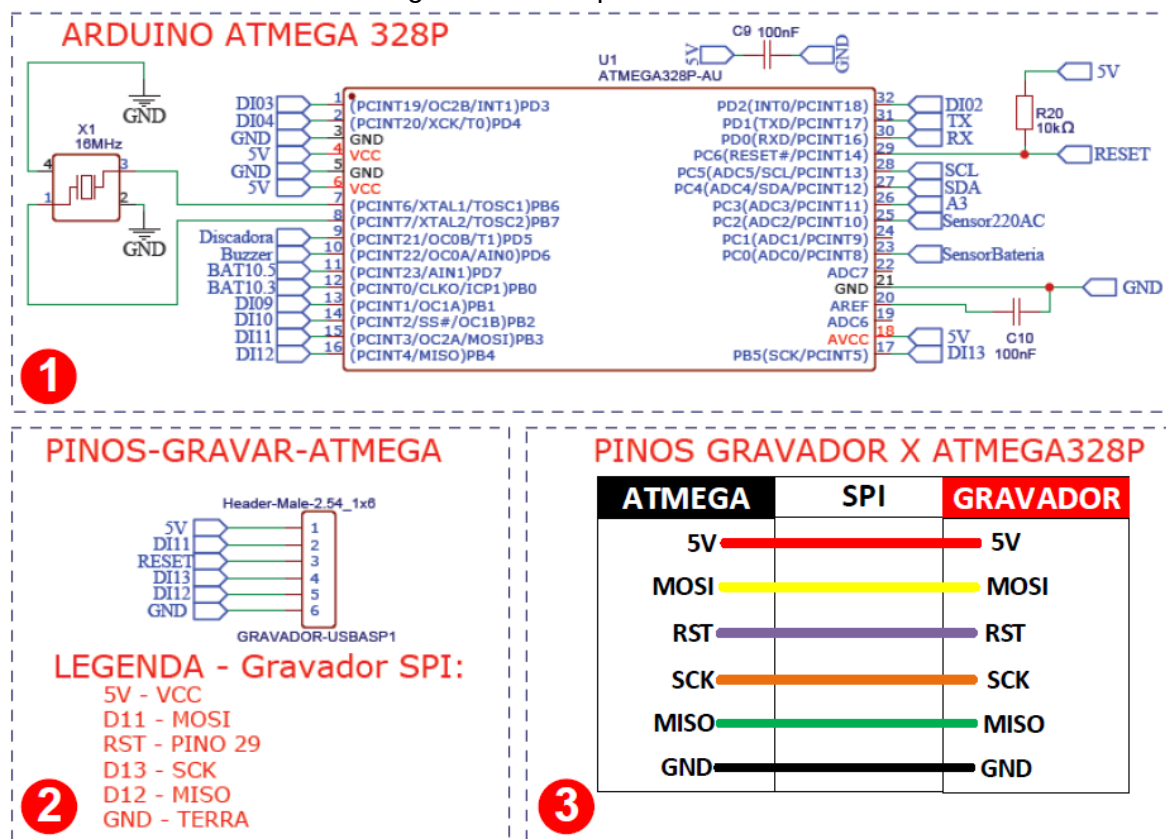
Este gravador é utilizado para programar microcontroladores AVR e opera no modo *in-circuit* (diretamente no circuito) através da interface SPI. Em um estudo, pesquisadores desenvolveram um Kit didático de baixo custo que inclui este gravador (Quintino et al., 2017). Na Figura 12, é apresentada uma foto do gravador. Um dos lados é conectado à porta USB do PC, enquanto os outros seis fios são conectados diretamente aos pinos do microcontrolador Atmega328P, como mostrado no esquema elétrico da Figura 13. Este esquema contém o hardware necessário para a programação do microcontrolador Atmega328P. Para obter mais informações sobre o gravador USBASP, consulte o manual disponível em Arduino e Cia, 2013.

Figura 12 – Gravador USBASP



Fonte: Arduino e Cia, 2013.

Figura 13 — Esquema elétrico



Fonte: Elaboração própria, 2024.

1) Esquema eletrônico das pinagens de gravação. 2) Pinos para conectar o gravador. 3) Ligação entre o gravador USBASP x Microcontrolador Atmega328P.

7.4.3 Sensor AC Voltage LM358

O sensor representado na Figura 14 é capaz de medir valores de tensão AC de 0 a 1000 V. Utilizando o Transformador ZMTP101B, ao ser energizado na entrada, converte a tensão de saída do secundário para um valor reduzido. Quando alimentado com 127 V AC, gera uma saída de 0,5 V AC de pico a pico. Por outro lado, ao ser alimentado com 220 V AC, produz uma saída de 1 V AC de pico a pico. Este projeto encontra-se detalhadamente descrito e disponível de forma aberta em Hareendran, 2022.

Figura 14 — Sensor de Tensão AC 0 a 250V Voltímetro ZMPT101B



Fonte: Hareendran, 2022.

Este sinal de baixa amplitude é condicionado passando por um filtro ativo com dois amplificadores operacional LM358 da empresa Texas Instruments e, em seguida, é enviado ao processador dentro da faixa de valores TTL (Hareendran, 2022). Dessa forma, é possível calcular o valor RMS da tensão da rede dentro do microcontrolador Atmega328P. A equação abaixo, representada como eq. 1, descreve a fórmula da tensão RMS, utilizada na biblioteca Emolib.h (Energy Monitor Library). Esta equação define que o valor da Tensão RMS é obtido pela raiz quadrada da média da soma das amostras dos valores ao longo de um ciclo completo (Período) (GLYN, 2019).

$$V_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} v[n]^2}$$

(1)

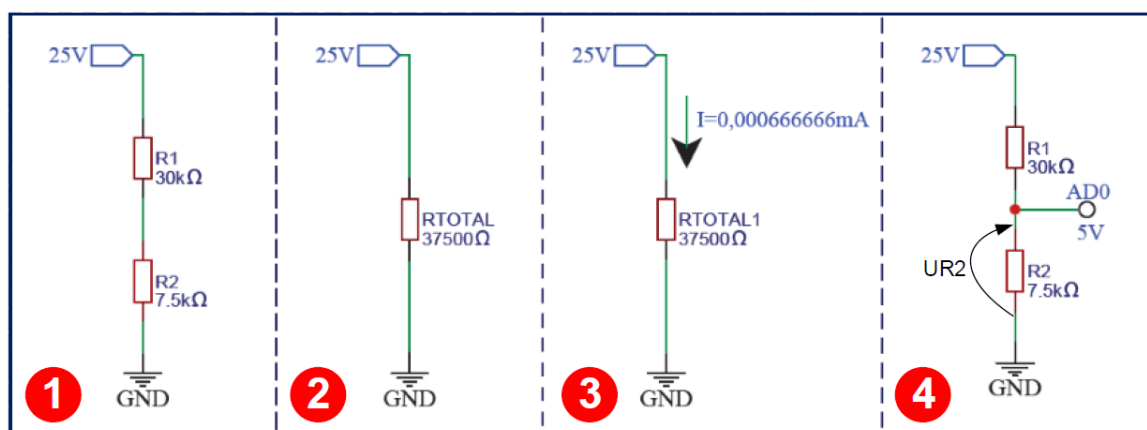
Tensão RMS da rede de 0 à 220 V AC.
Fonte: TrystanLea, GitHub, 2024.

Com base nos estudos realizados sobre este sensor, foi possível integrá-lo completamente na placa on-board (SMD) do Nobreak Inteligente. Após a conclusão do hardware, o código foi implementado utilizando a biblioteca Emonlib.h (Energy Monitor Library), disponível em Hudson (2015). Para incluir esta biblioteca em seu projeto, basta baixar o arquivo zip e adicioná-lo ao gerenciador de bibliotecas do ambiente Arduino.

7.4.4 Sensor DC de 0 à 25Vdc

Este é um sensor simples composto apenas por dois resistores em série, como demonstrado no circuito do sensor DC na Figura 15. Para medir de 0 a 25Vdc, essa configuração cria um divisor de tensão, no qual a saída da leitura do conversor analógico-digital (AD) do microcontrolador é projetada de modo que, ao aplicar 25 Vdc na entrada do sensor, a saída será, no máximo, 5Vdc. No contexto deste projeto, a fonte de alimentação fornece o valor de 14,4 V CC, o que está dentro da faixa suportada de até no máximo 25Vdc. Conforme ilustrado na Figura 16 e nos cálculos realizados, é possível compreender o valor da saída de 5 V CC.

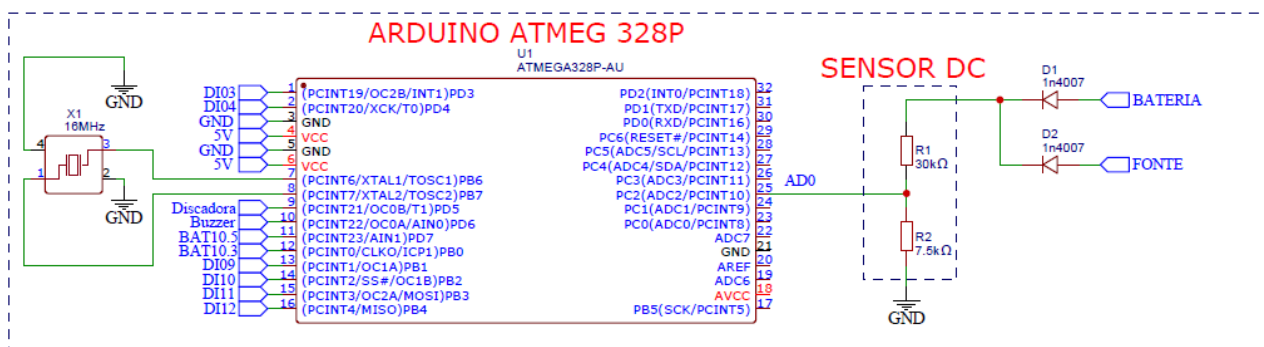
Figura 15 — Esquema Eletrônico do Sensor DC



Fonte: Elaboração própria, 2024.

- 1) Serie do divisor. 2) Resistência total do circuito. 3) Valor da Corrente do circuito. 4) Valor da tensão de 5 V CC acima do R2.

Figura 16 — Sequencia de cálculo do sensor DC na condição máxima de 25 V CC na entrada



Fonte: Elaboração própria, 2024.

Abaixo segue os detalhes dos cálculos de cada um dos quadros da Figura acima.

- a) Dois resistores em série $R1 = 30K$ e $R2 = 7.5K$.
- b) Resistência Total $\Rightarrow R_{total} = R1 + R2 = (30 \times 10^3) + (7,5 \times 10^3) = 37500 \text{ Ohm}$
- c) Corrente do Circuito $\Rightarrow I = U / R_{total} = 25 / 37500 = 0,000666666 \text{ A}$
- d) Tensão no $R2 \Rightarrow UR2 = I \times R2 = 0,000666666 \times 7500 = 5 \text{ V CC}$

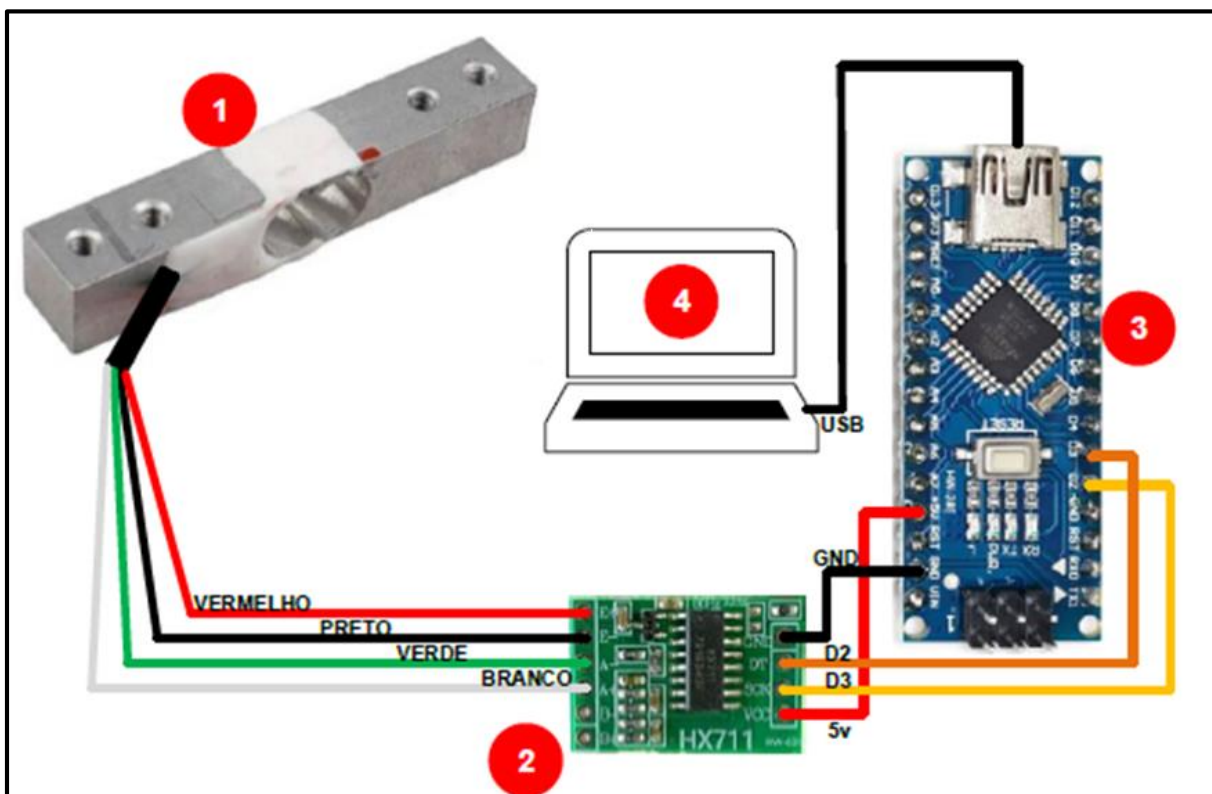
Com o valor garantido de no máximo 5 V CC na entrada AD0 do Atmega328p, foi possível implementar a leitura do sensor na IDE do Arduino. Considerando que a resolução do conversor analógico-digital do Atmega328P é de 10 bits, a resolução do sensor será de 0,00489V (5V / 1023). Abaixo segue a eq. (2), aplicada no microcontrolador, para calcular o valor DC da fonte/bateria.

$$TensaoDc = \left(\frac{LeituraAD0}{594.696} \right) + 0.65 \quad (2)$$

7.4.5 Sensor da Balança – Célula de Carga 3 kg – Amplificador hx711

Para controlar a variável peso da ração, foi utilizada uma célula de carga, que funciona como a balança responsável por pesar a ração. Como havia entradas disponíveis no microcontrolador Atmega328P do nobreak, decidiu-se implementar a balança da ração diretamente neste microcontrolador. Para testar a célula de carga, foi montado inicialmente o circuito conforme mostrado na Figura 17, que inclui um Arduino Nano, um CI HX711 (Amplificador de célula de carga), uma célula de carga de 3 kg e um PC com USB.

Figura 17 — Resultado da Balança e esquema do circuito da célula de carga para teste, calibração e validação da balança.

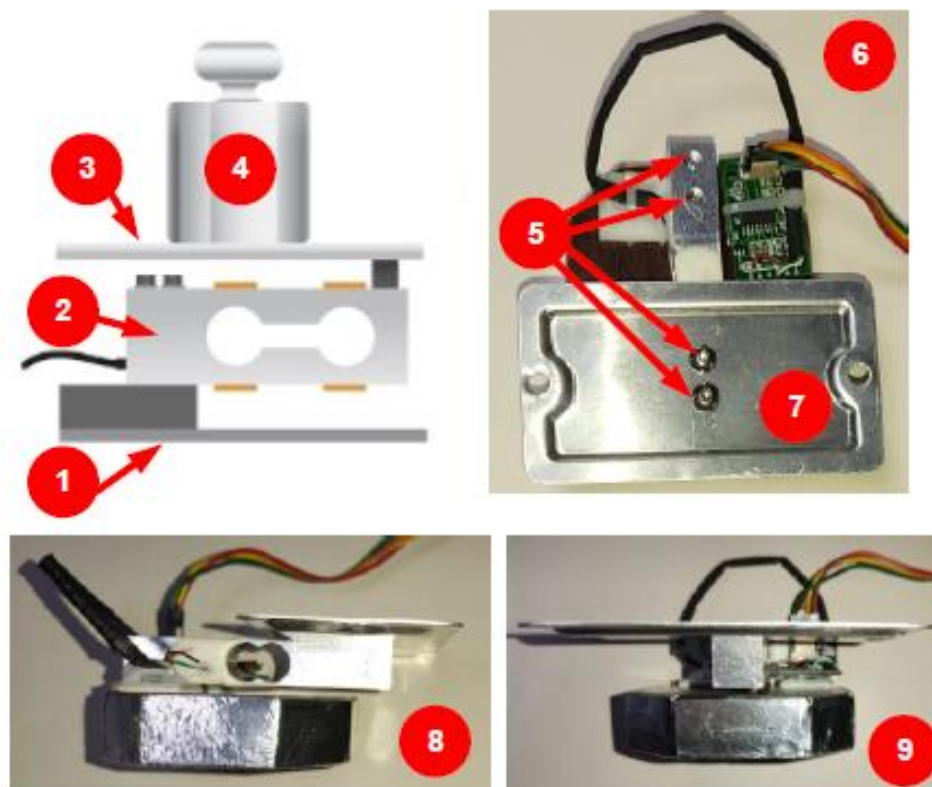


Fonte: Elaboração própria, 2024.

- 1) Célula de carga de 3kg. 2) CI Amplificador de sinal de célula de carga. 3) Arduino Nano – Atmega 328P. 4) PC com IDE Arduino e USB.

A estrutura mecânica da balança está detalhada e pode ser observada na Figura 18. Ela é composta por 4 partes distintas: 1) A base fixa inferior. 2) A célula de carga. 3) A base superior (prato). 4) Objeto a ser pesado (Ração). Todas essas partes são fixadas por quatro parafusos M4, utilizando os quatro furos com roscas M4 no corpo da célula de carga. Na parte central da balança, a célula de carga recebe o peso, causando uma deformação mecânica que resulta em uma variação na resistência elétrica. Este pequeno sinal elétrico é amplificado pelo CI HX711 e lido no Atmega328P.

Figura 18 — Detalhamento da Estrutura Física da balança da ração



Fonte: Elaboração própria, 2024.

- 1) Base fixa de baixo da balança. 2) Célula de carga de 3 Kg. 3) Base da parte de cima da balança (O prato). 4) Representa o objeto a ser medido (Peso). 5) Total de 4 parafusos M4 x 20mm para fixação da base de baixo e fixação da parte de cima (prato). 7) Vista de cima da balança. 8) Vista lateral da balança. 9) Vista frontal da balança.

Antes de montar o código da balança, foi necessário obter um peso padrão para garantir uma referência precisa. Para este projeto, foi utilizado um peso padrão de 200 gramas, que foi pesado e validado em uma balança. A Figura 19 apresenta a evidência da pesagem e validação do peso padrão de 200 gramas em uma balança industrial Amira.

Figura 19 — Amostra de peso padrão de 200 gramas usado para calibrar a balança da Ração



Fonte: Elaboração própria, 2024.

Conforme o programa mostrado na FIGURA 20, é apresentado o código do programa da balança, baseado na biblioteca (HX711.h). Na linha 14 do código, é possível identificar a função com o argumento `scale.set_scale(630811)`. Este valor numérico é utilizado para calibrar a balança. Para cada nova balança, esse valor será diferente, uma vez que cada equipamento possui seu próprio peso na célula de carga e no prato da balança. Este código está pronto e calibrado para medir o peso da ração. Se você deseja reproduzir esta balança, deve seguir os seguintes passos:

A) Grave o microcontrolador com o argumento `scale.set_scale(630811)`. Vazio, desta forma a função ficará assim: `scale.set_scale( )` e ao gravar o microcontrolador, a balança estará descalibrada e imprimirá valores aleatórios, o que é normal.

B) Coloque o peso padrão de 200 gramas sobre o prato. Neste momento, a balança ainda imprimirá valores desconhecidos.

C) Capture algumas amostras do monitor serial (quanto mais amostras, melhor).

Neste caso, 15 amostras forneceram uma precisão satisfatória. Calcule a média dos 15 valores e divida o resultado pela massa do peso padrão que você está utilizando. Neste caso, foram usados 0,2 kg (200 gramas).

Abaixo está a eq. (3) utilizada para determinar o valor correto de calibração da balança:

$$\text{Valor de calibração} = \left(\frac{\text{Média dos 15 valores impressos no monitor serial}}{0,2} \right) \quad (3)$$

Figura 20 — Firmware de teste, calibração e validação da balança

```

CelulaCarga$
1 // INCLUSÃO DE BIBLIOTECAS
2 #include <HX711.h>
3 // DEFINIÇÕES DE PINOS
4 #define pinDT 2
5 #define pinSCK 3
6 // INSTANCIANDO OBJETOS
7 HX711 scale;
8 // DECLARAÇÃO DE VARIÁVEIS
9 float medida = 0;
10 void setup()
11 {
12   Serial.begin(57600);
13   scale.begin(pinDT, pinSCK); // CONFIGURANDO OS PINOS DA BALANÇA
14   scale.set_scale(630811); // LIMPANDO O VALOR DA ESCALA
15   delay(2000);
16   scale.tare(); // ZERANDO A BALANÇA PARA DESCONSIDERAR A MASSA DA ESTRUTURA
17   Serial.println("Balança Zerada");
18 }
19 void loop() {
20   medida = scale.get_units(9); // SALVANDO NA VARIÁVEL O VALOR DA MÉDIA DE 5 MEDIDAS
21   Serial.println(medida, 3); // ENVIANDO PARA MONITOR SERIAL A MEDIDA COM 3 CASAS DECIMAIS
22   scale.power_down(); // DESLIGANDO O SENSOR
23   delay(1000); // AGUARDA 5 SEGUNDOS
24   scale.power_up(); // LIGANDO O SENSOR
25 }

```

Fonte: Elaboração própria, 2024.

Após a construção, treinamento e calibragem do dispositivo, ele foi colocado à teste em diferentes condições. Os resultados das simulações são apresentados na seção Resultados.

7.5 AUTONOMIA DOS SISTEMA DE BATERIA E CARREGAMENTO

Foi adotada uma bateria estacionária de 36Ah para o sistema. O fabricante recomenda que ela seja carregada com uma fonte que forneça, no máximo, 20% do valor em Ah, ou seja, até 7,2 A. A Figura 21 detalha o modelo DF500 da bateria da marca Freedom, destacando em amarelo os dados utilizados para dimensionar a autonomia. O fabricante sugere considerar como valor nominal a linha de 20h para cálculos de autonomia. Portanto, nesse caso, a eq. (4) abaixo resulta no valor em A que pode ser gasto por hora para uma autonomia de 20 horas.

$$\text{Corrente por hora} = \frac{\text{Capacidade da bateria (Ah)}}{\text{Tempo de descarga (horas)}} \quad (4)$$

Assim,

$$\text{Corrente por hora} = \frac{36 \text{ Ah}}{20 \text{ h}} = 1,8 \text{ A/h} \quad (4')$$

Figura 21 — DF500 - modelo de bateria estacionária adotada no circuito

PRINCIPAIS CARACTERÍSTICAS ELÉTRICAS						
MODELOS		DF300	DF500	DF700	DF1000	DF1500
Capacidade @25°C (Ah)	10 h	24	30	41	54	76
	20 h	26	36	45	60	80
	100 h	30	40	50	70	93

VALORES PARA O DIMENSIONAMENTO

Fonte: Fabricante Freedom

Com essa informação, foi possível dimensionar a autonomia do sistema do alimentador. Os circuitos eletrônicos considerando a placa do nobreak, a placa da câmera e a placa do *Datalogger* têm o consumo de corrente de 400 mA. Já na estrutura elétrica, tem-se a lâmpada de 5 watts que é ligada apenas na hora do almoço e jantar, desta forma, a Lâmpada = $5W / 127V = 40 \text{ mA}$. O motoredutor da ração consome a corrente de 7 A, porém é ligado apenas 2 vezes ao dia por 0,3 segundos cada vez, totalizando 0,6 segundos. A válvula solenoide da água que consome 100 mA e só liga para encher o pote de água. A Tabela 2 a seguir ilustra o consumo do sistema.

Tabela 2 — Consumo dos equipamentos eletroeletrônicos

Item	Descrição	Consumo	Tempo ligado
1	Todos placas eletrônicas	400 mA	100%
2	Motor da ração	7 A	0,6 seg / dia
3	Lâmpada	40 mA	2 horas / dia
4	Solenóide da água	100 mA	20 segundos / dia

Fonte: Elaboração própria, 2024.

Considerando a Tabela 2, em caso de falta de energia da tomada, a autonomia da bateria entrará em ação. Nesse caso o que mais impacta a duração da carga na bateria são os circuitos das placas eletrônicas que ficam ligados a todo tempo. A estrutura elétrica usa pouco a carga da bateria, com isso, aumenta o tempo da carga. A corrente do circuito do alimentador está entre a autonomia de 20 e 100 horas. Considerando o meio desses valores, tem-se um total de 60 horas de autonomia. Esse valor já apresenta uma boa margem de segurança. Portanto, está sendo adotado 48 horas de autonomia para esse projeto, com uma folga bem mais tranquila.

7.6 MICROCONTROLADOR ESP32CAM

O microcontrolador ESP32-CAM é um dos modelos da família ESP32, renomada por sua versatilidade. Equipado com uma câmera OV2640 de 2MP e um slot para cartão MicroSD. Na Figura 22, é possível visualizar o design compacto e funcional deste módulo. Com o ESP32-CAM, é viável criar projetos de monitoramento, captura de vídeo e imagens, além de enviar esses dados pela internet. Além disso, as imagens podem ser armazenadas diretamente no cartão MicroSD para visualização posterior. Em ambientes com pouca luminosidade, o LED flash integrado proporciona uma melhor visualização da cena. As características detalhadas desse dispositivo podem ser verificados em (AI-THINKER, 2017) e nos itens abaixo.

- Processador: Xtensa® Dual-Core 32-bit LX6;
- Memória Flash programável: 4 MB (dos quais 0,9 Mb são usados pelo bootloader);
- Memória RAM: 520 KBytes;
- Memória ROM: 448 KBytes;

- Clock máximo: 240 MHz;
- Pinos Digitais GPIO: 11 (todos com PWM);
- Resolução do PWM: até 16 bits (ajustável via código);
- Wireless 802.11 b/g/n - 2.4GHz (antena integrada);
- Modos de operação: Access Point / Estação / Access Point + Estação;
- Bluetooth Low Energy padrão 4.2 integrado;
- Slot para câmera OV2640 de 2 mega pixel;
- Slot de micro sd card;

Figura 22 — ESP32CAM



Fonte: <https://github.com/raphaelbs/esp32-cam-ai-thinker/blob/master/assets/ESP32-CAM_Product_Specification.pdf>, Acessado em: 02/03/2024

7.7 PLATAFORMA *EDGE IMPULSE* – TREINAMENTO DA REDE NEURAL

O objetivo com essa tecnologia é preencher a lacuna de que animais diferentes podem comer a ração um do outro. Outros estudos tentaram resolver esse problema usando um chaveiro de RFID como coleira para autenticar o animal de estimação para liberar a ração. Nesta dissertação, no entanto, será utilizada a ESPCAM com a câmera, que usa um algoritmo de detecção de imagem para classificar e diferenciar uma cachorra de um gato. O dispositivo só irá dispensar a ração se for a cachorra

Dolly. Na Figura 23, pode ser observada a cachorra que participou do estudo.

Figura 23 — Dolly a cachorra utilizada no treinamento



Fonte: Elaboração própria, 2024.

Essa classificação é realizada com algoritmo criado na plataforma *Edge Impulse*. Primeiramente, o usuário deve se cadastrar na plataforma, criar um projeto, escolher um nome do projeto. Nela, é realizado o treinamento da rede neural seguindo os passos, conforme abaixo na FIGURA 24.

Figura 24 — Passos do treinamento da rede neural na plataforma *Edge Impulse*.

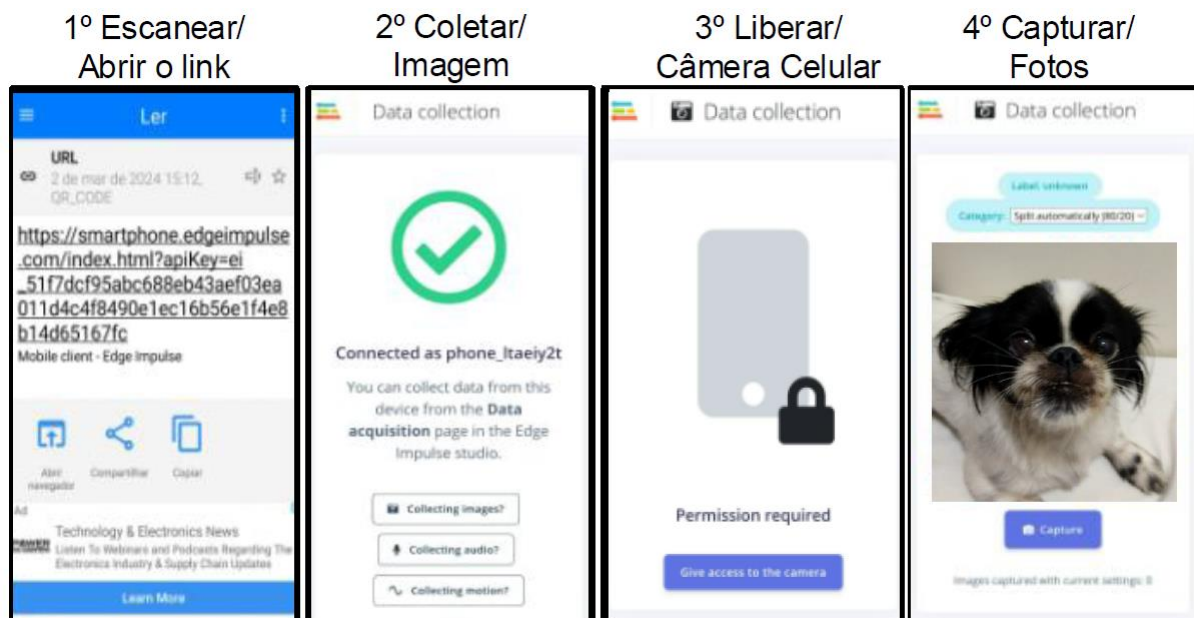


Fonte: Edge Impulse, 2024

7.7.1 Passo 1 - Edge Impulse (Coleta Imagens)

Aqui são inseridos os dados (fotos) do que deseja ser classificado. Pode ser carregado os dados (fotos) de três formas diferentes, através da leitura do Qrcode, que é a partir do celular, ou através do computador com uma webcam, ou através da placa de desenvolvimento, como *Raspberry* e ESP32CAM. Aqui devem ser coletadas várias fotos dos objetos a serem classificados e é necessário obter fotos em várias posições diferentes. Para esse treinamento, através da câmera do celular, foram realizadas a captura de 40 fotos para cada um dos dois objetos, da cachorra e do gato. Abaixo na Figura 25 são demonstrados os 4 passos para fazer a captura das fotos. Após a leitura do Qrcode pelo celular, abrirá um link. O procedimento deve ser: clicar e aguardar a sincronização do aparelho na plataforma, pressionar o botão coletar imagem, liberar acesso da câmera do celular e tirar as fotos. A partir desse momento, a cada clique no botão capture é salvo uma foto no projeto ativo no *Edge Impulse*.

Figura 25 — Sequencia de captura das fotos através do celular na plataforma *Edge Impulse*

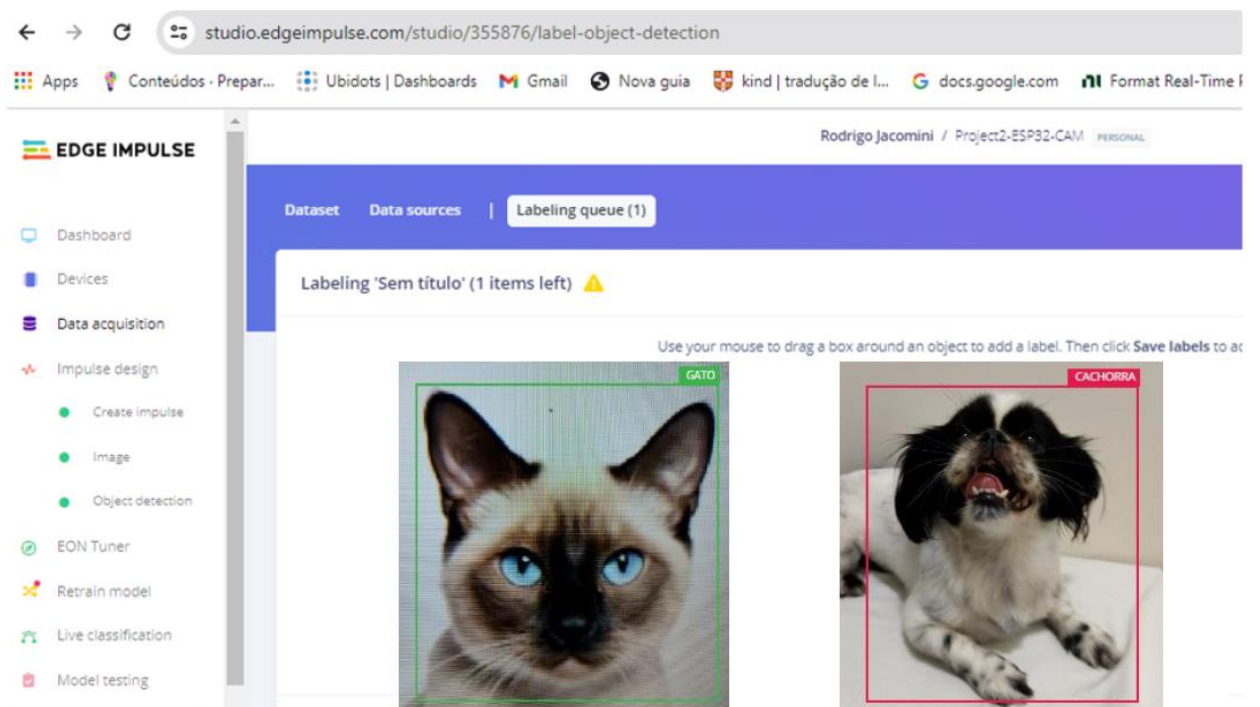


Fonte: Elaboração própria, 2024

7.7.2 Passo 2 - Edge Impulse (Inclusão de Lables).

Após todas as fotos serem salvas no seu novo projeto, iniciou-se a inclusão dos labes (rótulos) em cada uma delas. Por exemplo, nesse treinamento, foram retiradas 40 fotos da cachorra em várias posições diferentes. Em cada uma das fotos, deve ser incluído o *label* Cachorra Figura 26. Da mesma forma, foram incluídos o *label* Gato em cada foto da gata. Foi incluído um frame quadrado o mais próximo possível das características que identifiquem melhor o objeto. No exemplo do frame do gato, foi enquadrado a cabeça para melhor identificação na hora da treinamento e classificação. É fundamental assegurar que exista boa iluminação e plano de fundo limpo e organizado, porque trará maior nitidez nos dados coletados, consequentemente, um bom treinamento da rede e posteriormente uma boa classificação.

Figura 26 — Inclusão do frame quadrado e nome do *label* na plataforma *Edge Impulse*



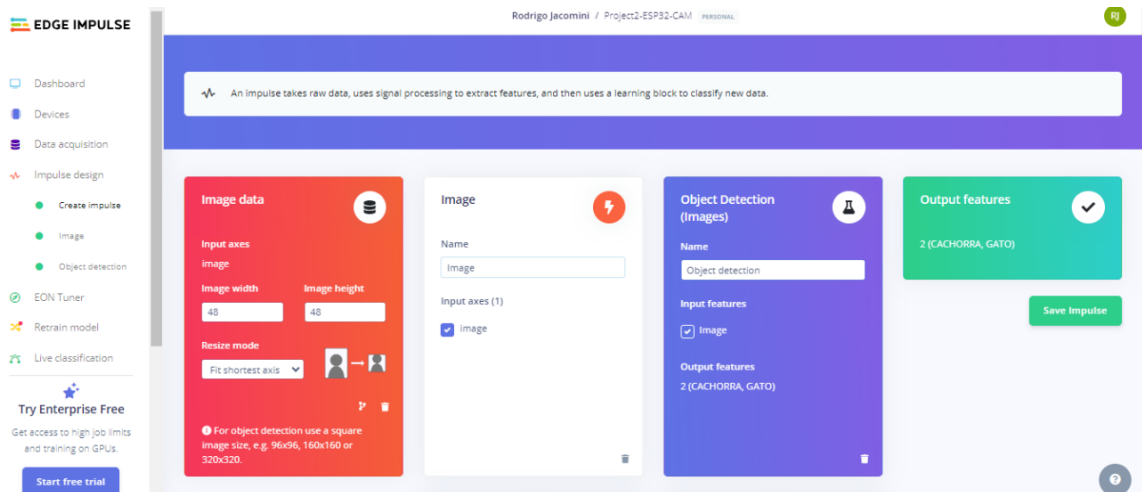
Fonte: Elaboração própria, 2024

7.7.3 Passo 3 - Edge Impulse (Impulse Design)

Nesse passo, foi selecionado o *Edge Impulse Design*. É a parte que deve ser configurado como o algoritmo irá trabalhar na classificação, por exemplo, se será

reconhecimento de voz ou classificação de objetos por visão computacional. Para esse treinamento, foi selecionado o treinamento com foto, que irá realizar detecção de objetos e como saída terão duas classes distintas, Gato e Cachorro (Figura 27).

Figura 27 — *Impulse Design* na plataforma *Edge Impulse* (Dados do Autor)



Fonte: Elaboração própria, 2024

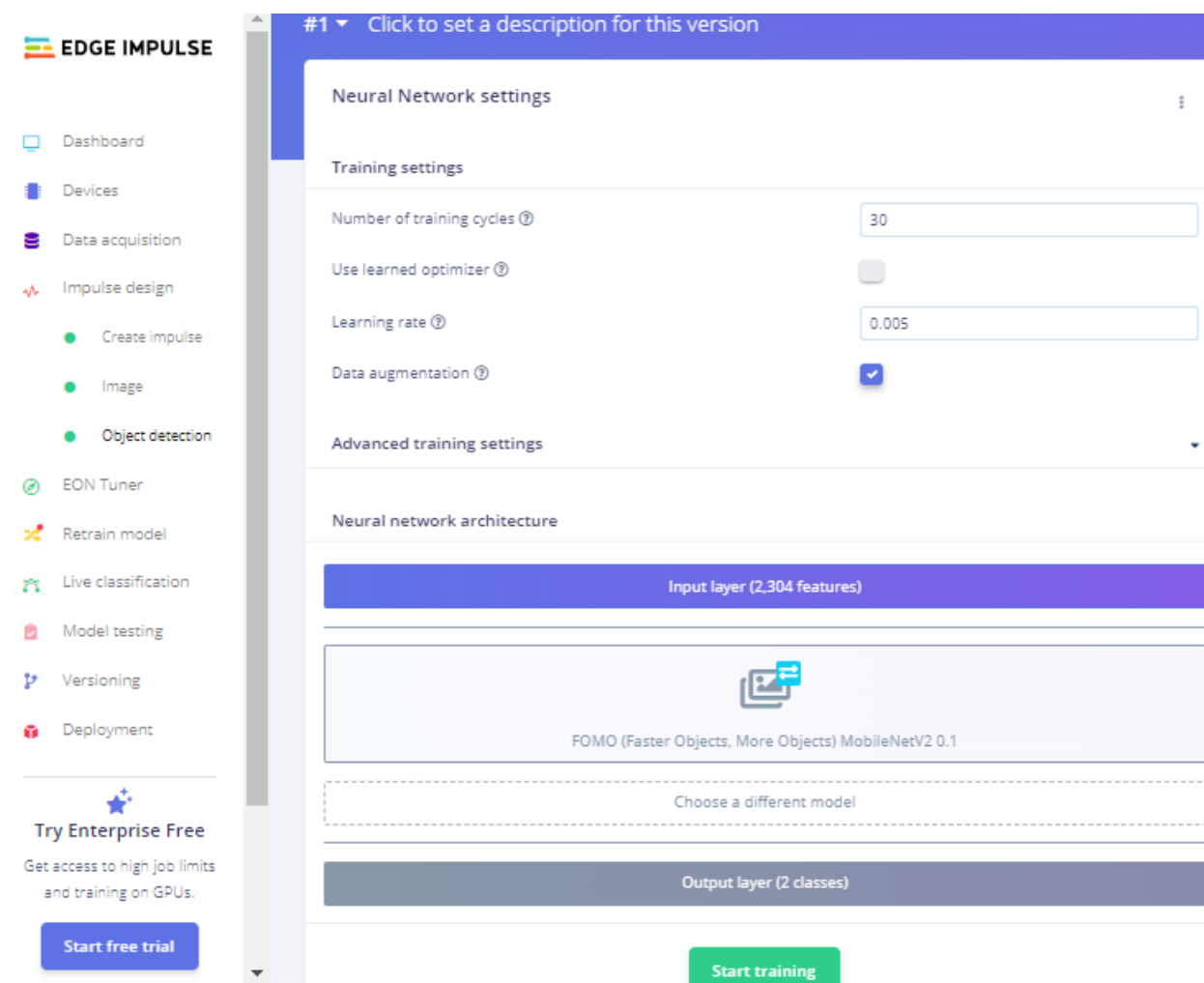
Nesse passo de ajuste da rede neural via *Edge Impulse*, é feita a seleção do algoritmo da rede neural para orquestrar o treinamento. Os algoritmos de redes neurais disponíveis na plataforma são:

- MobileNetV2 SSD FPN-Lite 320x320;
- FOMO (Faster Objects, More Objects) MobileNetV2 0.1
- YOLOv5

A rede neural selecionada foi a FOMO, baseada na rede MobileNetV2. A plataforma *Edge Impulse* esclarece que o algoritmo de rede neural FOMO é um novo algoritmo de aprendizado de máquina que traz a detecção de objetos para dispositivos altamente restritos. Ele permite contar objetos, encontrar a localização de objetos em uma imagem e rastrear vários objetos em tempo real usando até 30 vezes menos poder de processamento e memória do que MobileNet ou YOLOv5. Sendo assim, o FOMO é ideal para IA aplicada em microcontroladores. Na Figura 28, é possível verificar a configuração e parametrização/peso da rede neural antes de iniciar o treinamento da rede (Menezes, 2023). Um estudo afirmou que a maioria dos modelos de rede neural para detecção de objetos têm mais memória e poder de processamento do que os microcontroladores (Menezes, 2023). Já o modelo do tipo FOMO requer

apenas poucas centenas de KB de memória, com isso, torna promissora a aplicação de algoritmos de detecção em microcontroladores.

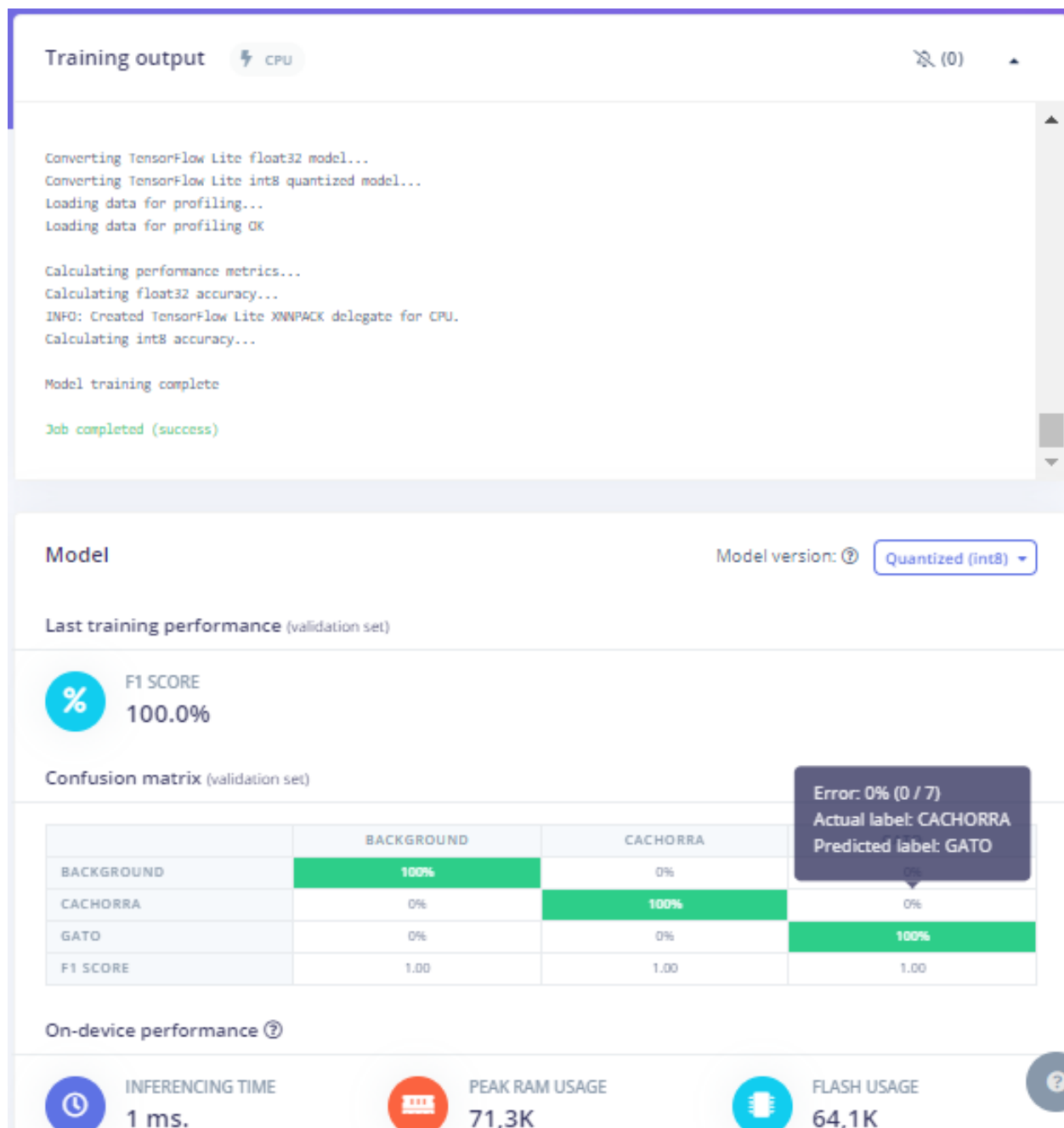
Figura 28 — Parametrização da rede neural - Plataforma *Edge Impulse*
(Dados do Autor)



Fonte: Elaboração própria, 2024

Após a parametrização, ao pressionar o botão 'Iniciar Treinamento', a rede neural começará a ser treinada. Na Figura 29, são apresentados os resultados do treinamento da rede neural, incluindo a tabela da matriz de confusão, bem como os requisitos mínimos de hardware para executar o processo de inferência em 1 milissegundo. Esse tempo inclui a captura da imagem, classificação e geração de saída.

Figura 29 — Resultado da rede neural já treinada - Plataforma *Edge Impulse*

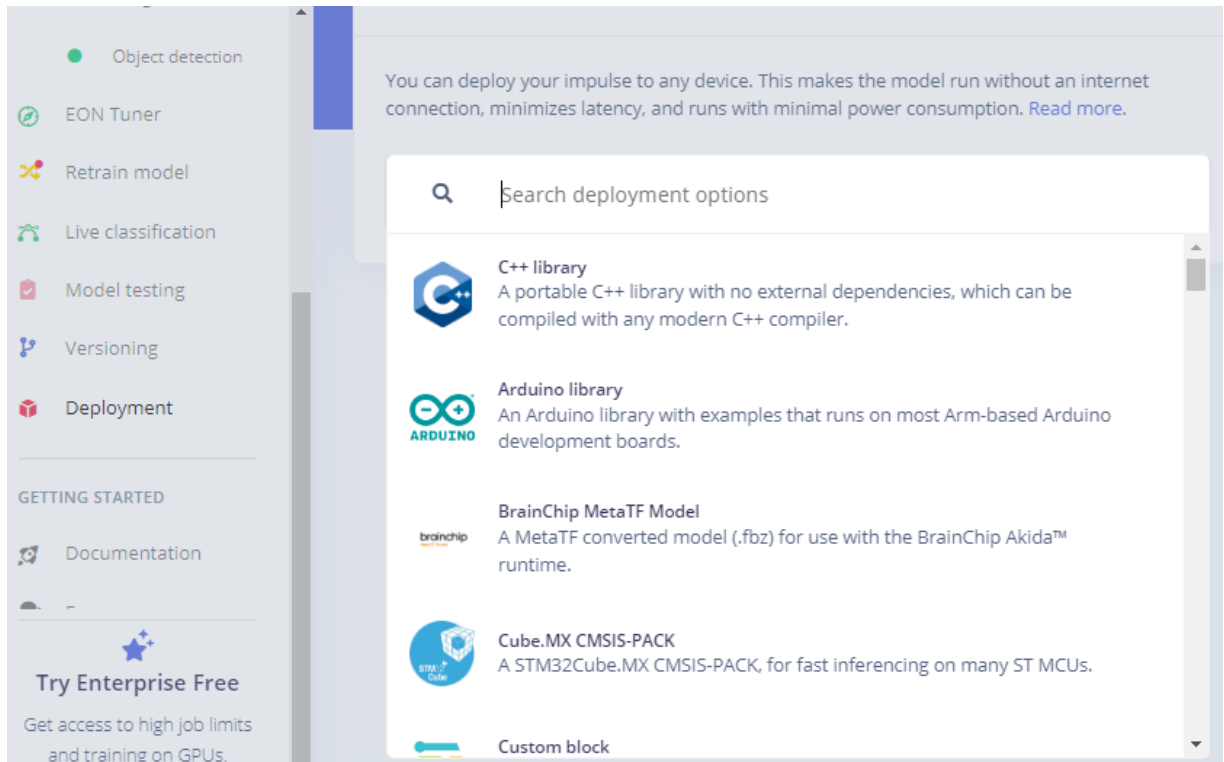


Fonte: Elaboração própria, 2024

7.7.4 Passo 4 - Edge Impulse (Deployment)

Nesse passo, pode-se implantar o impulso em qualquer dispositivo. Com a vantagem de o modelo funcionar sem conexão com a Internet, minimiza a latência e funciona com consumo mínimo de energia. Conforme a Figura 30 é possível escolher se vai rodar em ESP32, STM32, RASPIBARRY PI4, etc.

Figura 30 — Escolha de qual microcontrolador irá rodar o algoritmo



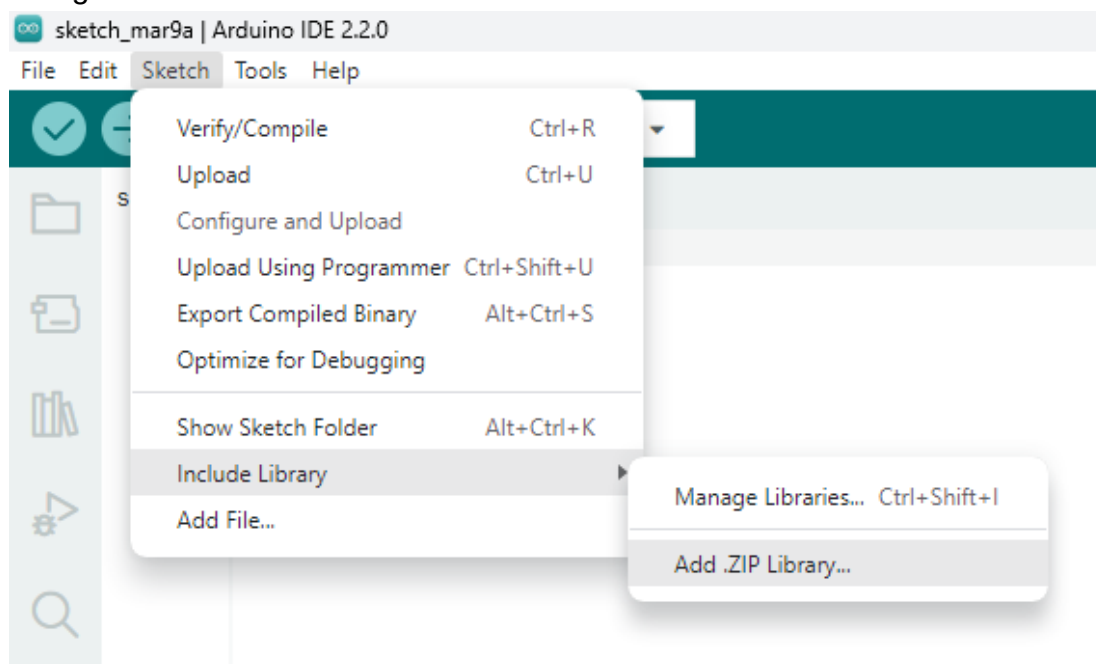
Fonte: Elaboração própria, 2024

Neste caso, para o treinamento destinado à placa ESP32CAM, foi selecionado o ambiente Arduino para desenvolver o algoritmo de classificação. Agora, basta pressionar o botão 'Build' (Construir) para gerar os códigos a serem executados na IDE do Arduino. Após pressionar o botão 'Build', os códigos, bibliotecas e algoritmos de classificação serão automaticamente compilados, criados e um arquivo zip será baixado para o computador.

7.7.5 Passo 5 - Edge Impulse (Run Model)

Esta etapa envolve a preparação da IDE Arduino para compilar, gravar o microcontrolador e executar o sistema de classificação com a ESP32-CAM utilizando o algoritmo de classificação de cães ou gatos. Após o download do arquivo .zip, ele deve ser incorporado à gestão de bibliotecas na IDE do Arduino. Na Figura 31, é demonstrado o processo a ser seguido na IDE do Arduino para incluir o arquivo compactado que foi baixado da plataforma *Edge Impulse*.

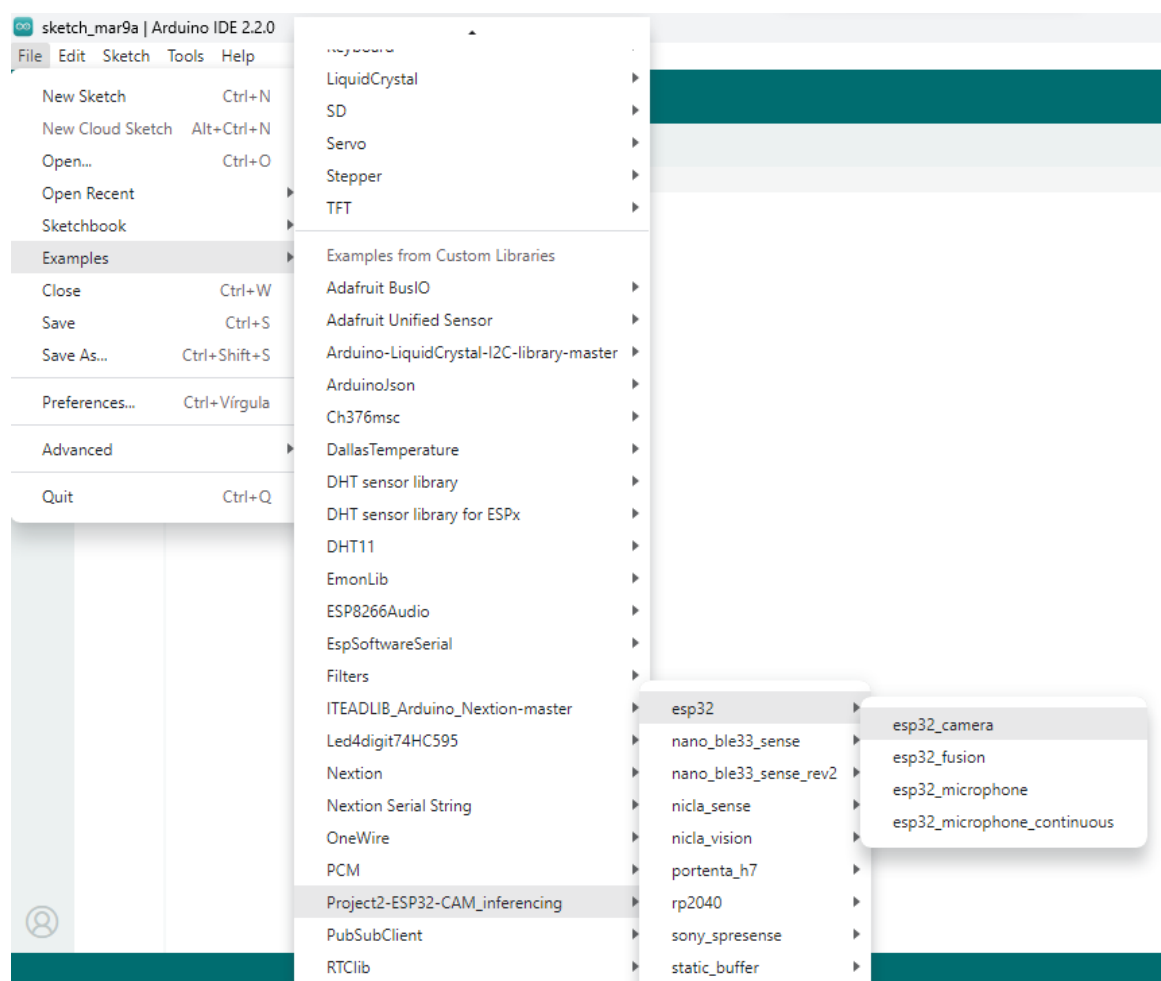
Figura 31 — Gerenciamento e inclusão da biblioteca na IDE Arduino



Fonte: Elaboração própria, 2024

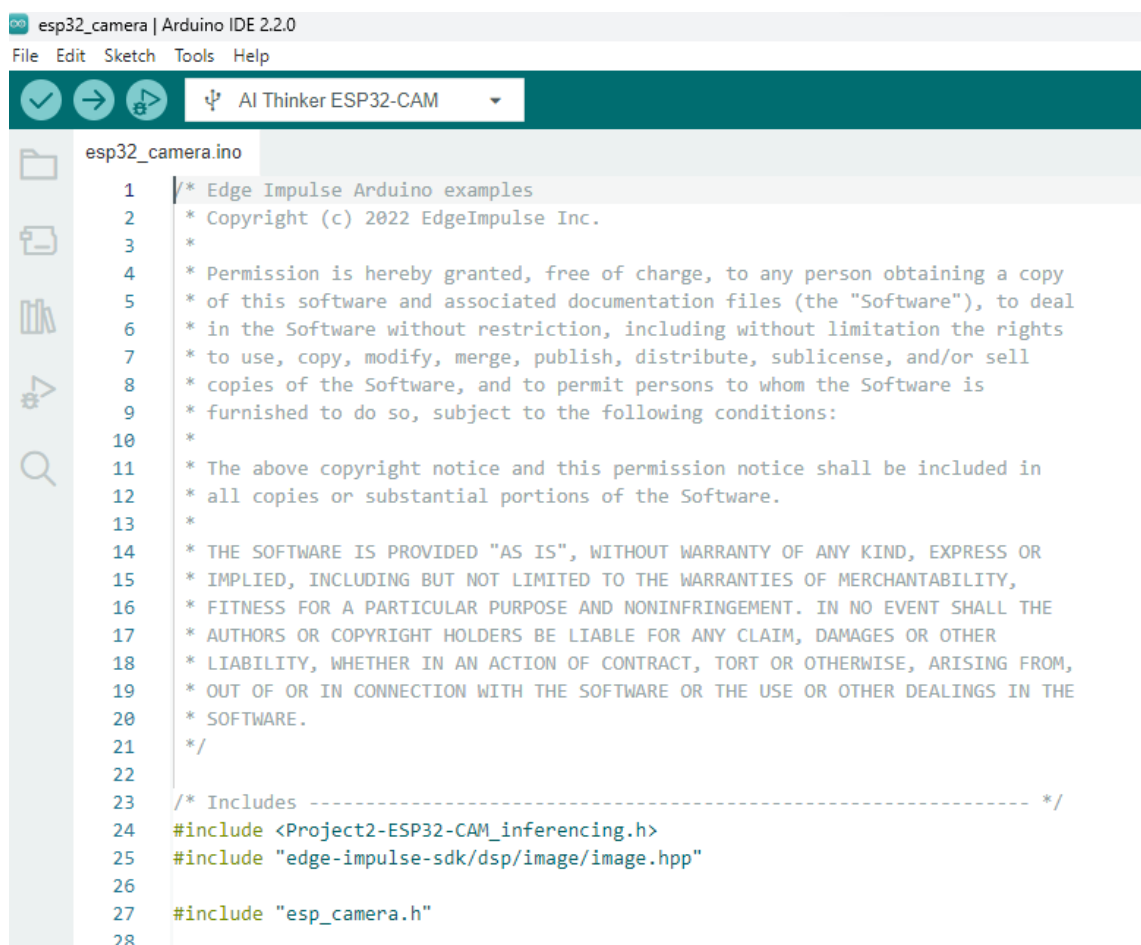
Com o arquivo zip da biblioteca adicionado, é possível, seguindo o caminho indicado na Figura 32, abrir o projeto 'esp32_camera'. Na Figura 33, é possível verificar o código aberto na IDE, que será carregado na ESPCAM. Este código já incorpora o algoritmo de classificação.

Figura 32 — Caminho do algoritmo do projeto classificação com esp32_câmera



Fonte: Elaboração própria, 2024

Figura 33 — Projeto esp32_câmera aberto na IDE Arduino



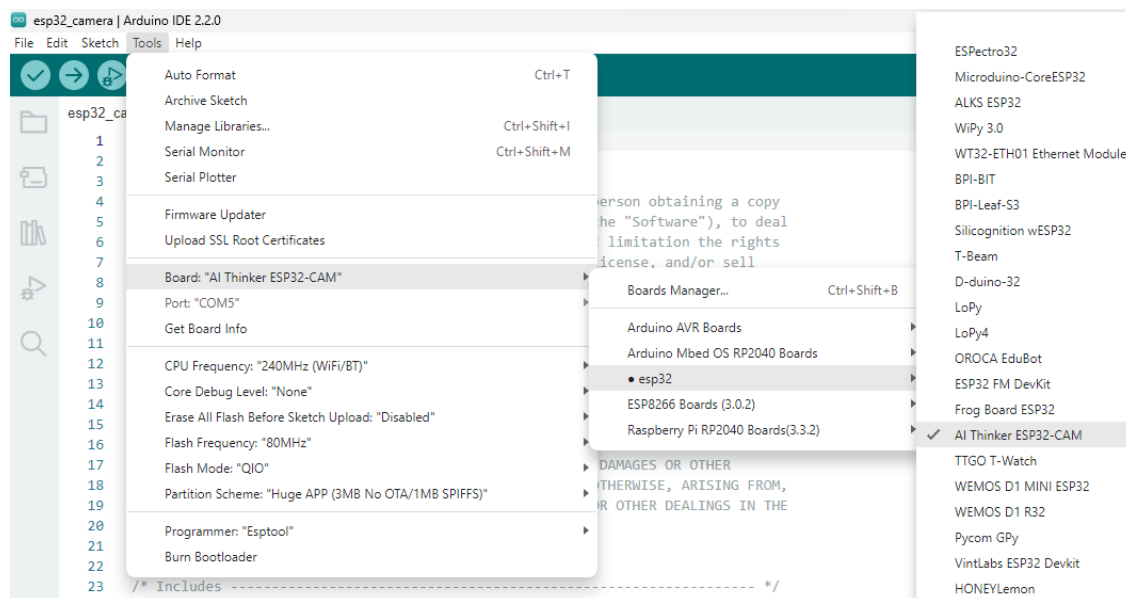
```
esp32_camera | Arduino IDE 2.2.0
File Edit Sketch Tools Help

esp32_camera.ino
1  /* Edge Impulse Arduino examples
2  * Copyright (c) 2022 EdgeImpulse Inc.
3  *
4  * Permission is hereby granted, free of charge, to any person obtaining a copy
5  * of this software and associated documentation files (the "Software"), to deal
6  * in the Software without restriction, including without limitation the rights
7  * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
8  * copies of the Software, and to permit persons to whom the Software is
9  * furnished to do so, subject to the following conditions:
10 *
11 * The above copyright notice and this permission notice shall be included in
12 * all copies or substantial portions of the Software.
13 *
14 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
15 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
16 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
17 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
18 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
19 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
20 * SOFTWARE.
21 */
22
23 /* Includes ----- */
24 #include <Project2-ESP32-CAM_inferencing.h>
25 #include "edge-impulse-sdk/dsp/image/image.hpp"
26
27 #include "esp_camera.h"
28
```

Fonte: Elaboração própria, 2024

Na Figura 34, é apresentado o processo de seleção da placa ESP32-CAM na IDE garantindo assim o ambiente correto para a execução no microcontrolador.

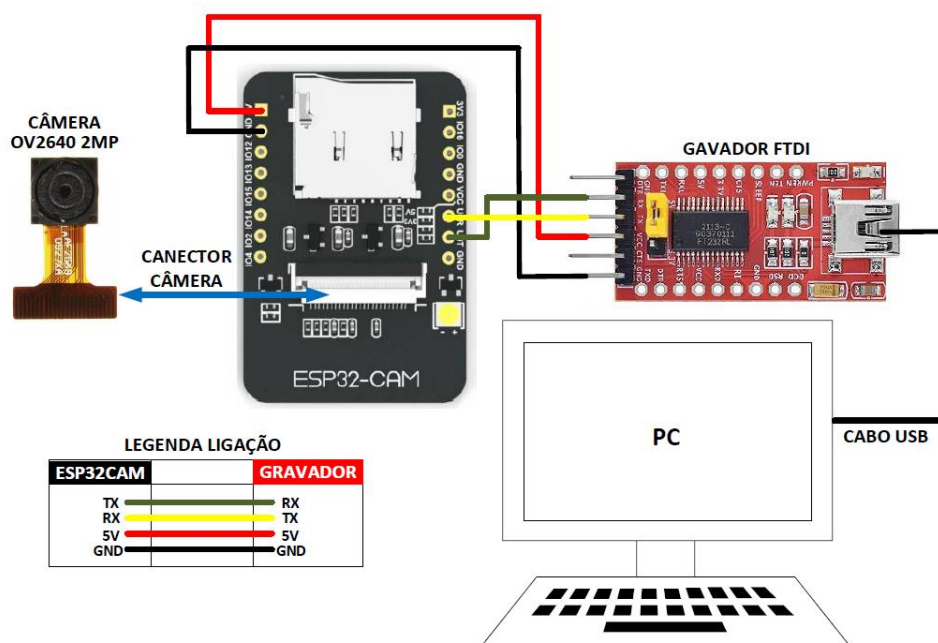
Figura 34 — Escolha da placa microcontrolador AI Thinker ESP32-CAM



Fonte: Elaboração própria, 2024

A Figura 35 abaixo mostra como devem ser conectados o gravador, a câmera e a ESP32-CAM à porta USB do computador para gravar o programa compilado na placa. Como a placa não possui um CI USB integrado, é necessário utilizar um gravador FTDI para gravar o microcontrolador. Na Figura, são apresentados o conjunto montado e os componentes utilizados no teste de classificação.

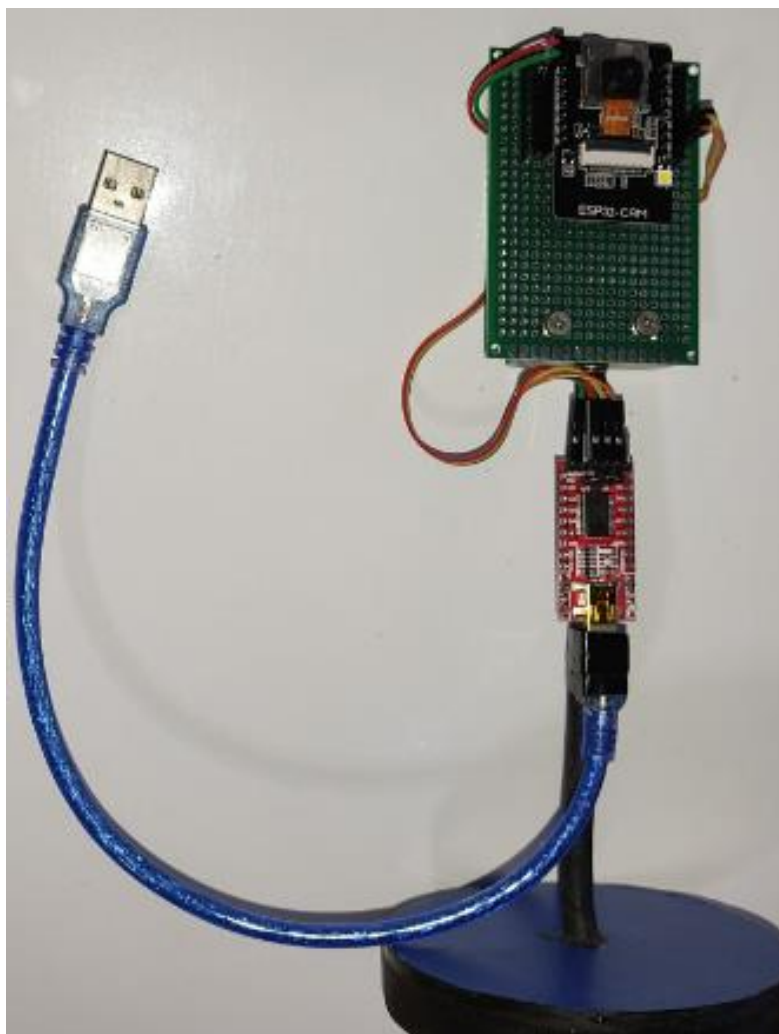
Figura 35 — Detalhes de ligação do gravador FTDI, ESP32-CAM, câmera OV2640 2MP e o PC



Fonte: Elaboração própria, 2024

Na Figura 36 abaixo, apresenta-se a configuração física do teste com a ESP32-CAM, incluindo uma base mecânica, conversor e cabo USB.

Figura 36 — Base montada para ESP32-CAM capturar as imagens



Fonte: Elaboração própria, 2024

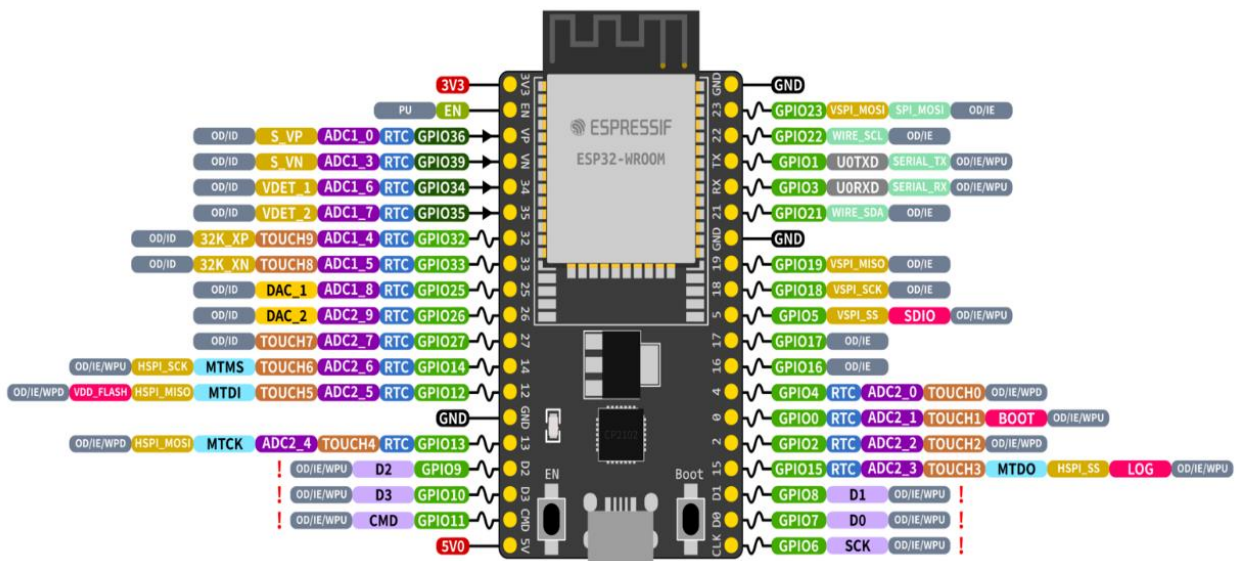
7.8 MICROCONTROLADOR ESP32-WROOM-32

O ESP32-WROOM-32 (Figura 37) é um módulo Wi-Fi + Bluetooth, que pode ser utilizado em várias aplicações, desde redes de sensores de baixa potência até as tarefas mais exigentes, como codificação voz, streaming de música e vídeos. Contém dois núcleos de CPU que podem ser controlados individualmente e opera na frequência entre 80 MHz a 240 MHz. O ESP32 tem inúmeros periféricos, como, sensor de toque capacitivo, interface de cartão SD, Wi-Fi, Bluetooth, SPI de alta velocidade, UART, I2S e I2C (Espressif, 2023). Abaixo segue as especificações:

- Processador: Xtensa 32-Bit LX6 Dual Core;
- Memória SRAM: 520Kb;

- Memória Flash Externa: 32-Bit de acesso e 4Mb;
- Tensão de Alimentação: 2,7 à 3,6 VDC;
- Corrente de consumo: 80mA (típica);
- Corrente de consumo: 500mA (máxima);
- WiFi: 802.11 b/g/n: 2.4 à 2.5 GHz;
- Segurança WiFi: WPA / WPA2 / WPA2-Enterprise / WPS;
- Bluetooth: 4.2 BR / EDR e BLE (Bluetooth Low Energy);
- Suporte: Arduino IDE, Lua e MicroPython, Código C básico;
- Interfaces: UART, SPI, SDIO, I2C, I2S, IR, PWM;
- Temperatura de trabalho: -40° à +85° C;

Figura 37 — Pinagem da ESP32-WROOM-32



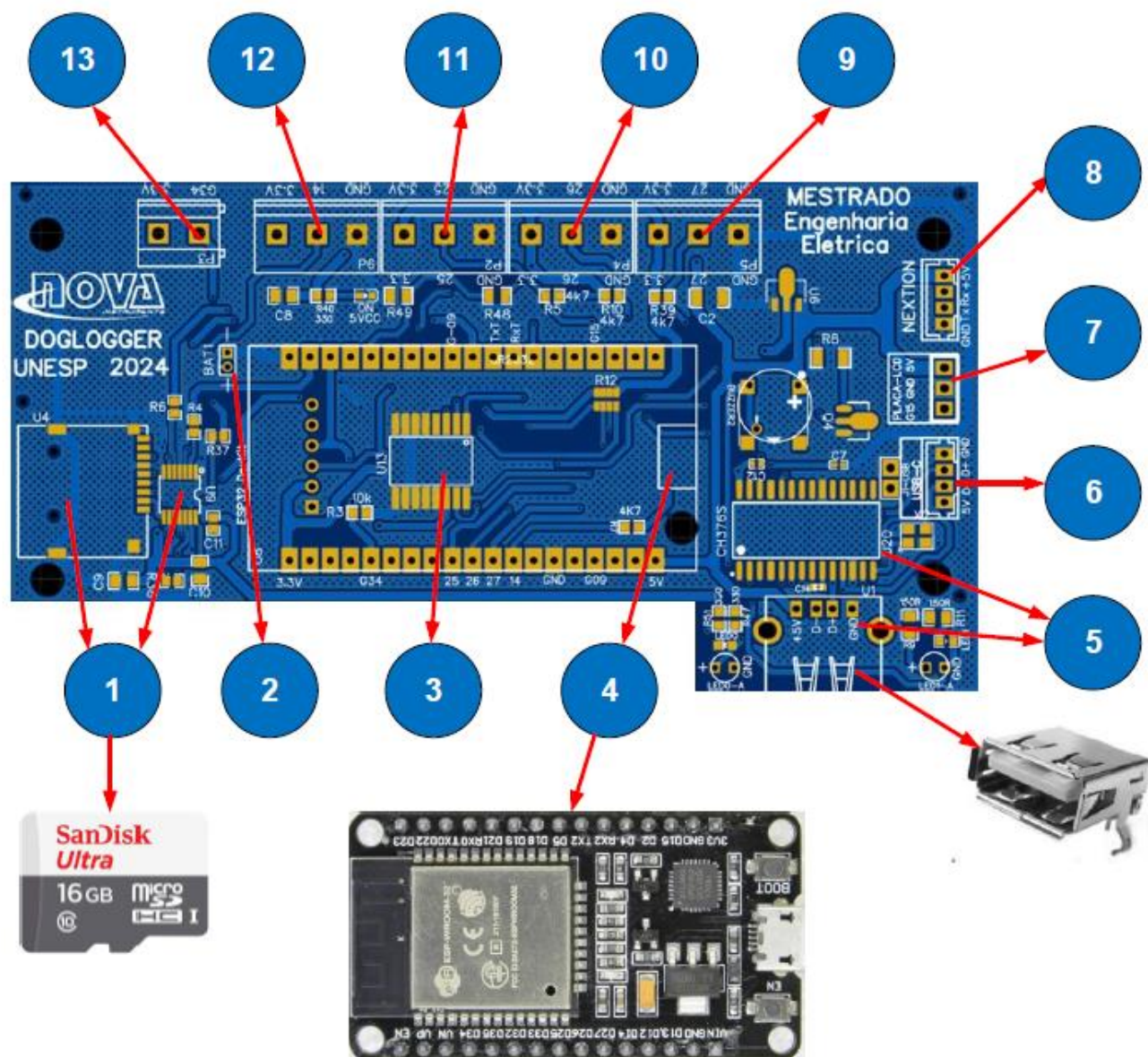
Fonte: Espressif, 2023.

7.9 SISTEMA IOT DO PROJETO

O sistema IoT será composto por três partes: 1 - Hardware do sistema; 2 - IDE de programação do sistema embarcado; 3 - Adoção do serviço/servidor de nuvem. Na Figura 38, apresenta-se o hardware completo responsável por controlar água, ração, luz, receber os valores de Peso da Ração, AC, DC do nobreak, receber a classificação da ESPCAM (Cachorro, Gato), gravar o *Datalogger* local (cartão SD) e, por fim, realizar a publicação na nuvem.

7.9.1 IoT parte 1 - Hardware do sistema

Figura 38 — Placa responsável pelo controle com a ESP32 do RTC



Fonte: Elaboração própria, 2024

- 1) CI-SMD, Slot de SMD e micro sd card. 2) Conector de bateria 2032 de 3,3V. 3) RTC - CI DS3231-SMD. 4) Soquete do Microcontrolador ESP32-WROOM-32 de 38 pinos. 5) CI-CH3765-SMD (Conversor USB-RS232). 6) Outra tomada USB. 7) Pinos de Alimentação GND, 5V e 1 pinos software serial. 8) Póde ligar display gráfico nextion – 4 pinos: GND, TX, RX e 5V. 9) Conector com 3V3, GPIO27 e GND. 10) Conector com 3V3, GPIO26 e GND. 11) Conector com 3V3, GPIO25 e GND. 12) Conector com 3V3, GPIO14 e GND. 13) Conector com 3V3, GPIO34 e GND.

Essa placa será responsável pelo controle com a ESP32 do RTC (Relógio de Tempo Real), receberá os valores de peso da ração, de tensão da tomada e da bateria do nobreak. Além disso, receberá a informação sobre se o cachorro foi reconhecido pela classificação do ESPCAM. Ao receber esses dados, o processador executará algumas condições para servir a ração:

- Primeiramente, verificará se a classificação da ESPCAM é o um cachorro.
- Em seguida, verificará se já foi servida alguma vez a ração do almoço ou do jantar.

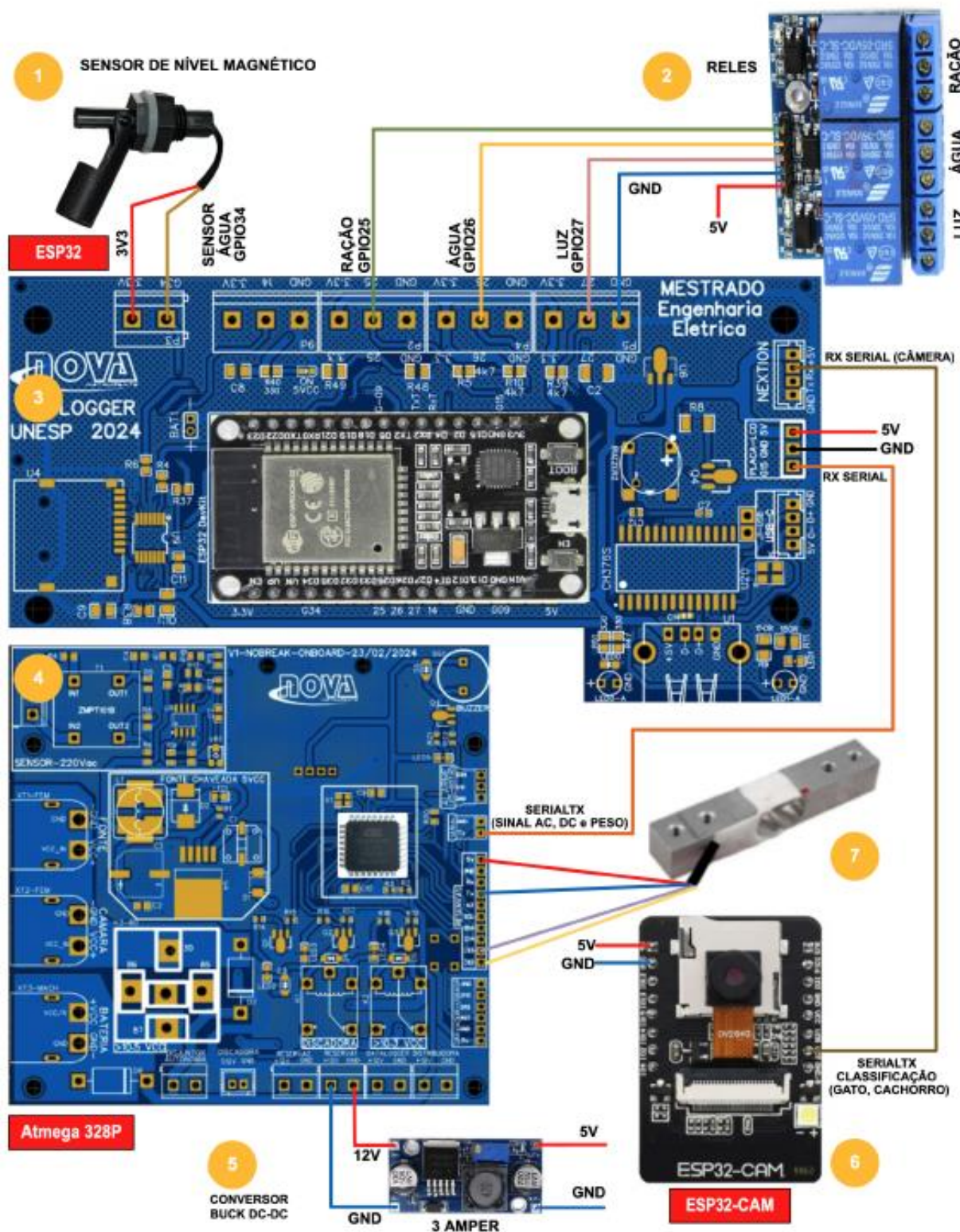
Por último, verificará o relógio para determinar se está dentro do horário de almoço ou jantar.

Toda vez que houver uma atividade de reconhecimento do cachorro ou do gato, será registrado no arquivo do *Datalogger* no cartão SD e esses dados também serão enviados para a nuvem. Da mesma forma, um registro será incluído quando a ração e a água forem servidas. No entanto, se não houver atividade dos animais, um novo registro será incluído no cartão SD após decorridos 5 minutos e na nuvem a cada 7 minutos. Os dados registrados incluirão:

- Número do Registro;
- Data;
- Hora;
- Nível da Tomada AC;
- Nível da Bateria DC;
- Serviu Ração;
- Serviu Água."

Na Figura 39, está demonstrada a interconexão entre todas as placas eletrônicas que compõem toda a inteligência do sistema de segurança de alimentação para cachorro. Esta Figura inclui a placa com câmera para classificação, o nobreak inteligente com sistemas de fonte e a placa IoT / *Datalogger*, além da boia de água.

Figura 39 — Esquema de ligação dos periféricos do sistema de controle



Fonte: Elaboração própria, 2024

- 1) Boia magnética da água 2) Relés 3) Placa do sistema de controle (PSC) 4) Placa de suprimento de energia Inteligente (SSEI) 5) Fonte Chaveada 6) ESP32CAM placa do sistema de identificação de Pets (SIP) 7) Sistema com célula de carga

7.9.2 IoT parte 2 - IDE de programação do sistema embarcado

O ESP32-WROOM foi programado no IDE Arduino. Devido à execução de várias atividades, o microcontrolador será programado para utilizar os dois núcleos internos. Na Figura 40, é possível verificar a divisão das funções em cada núcleo. O loop principal será executado no core 1 e terá as seguintes funções:

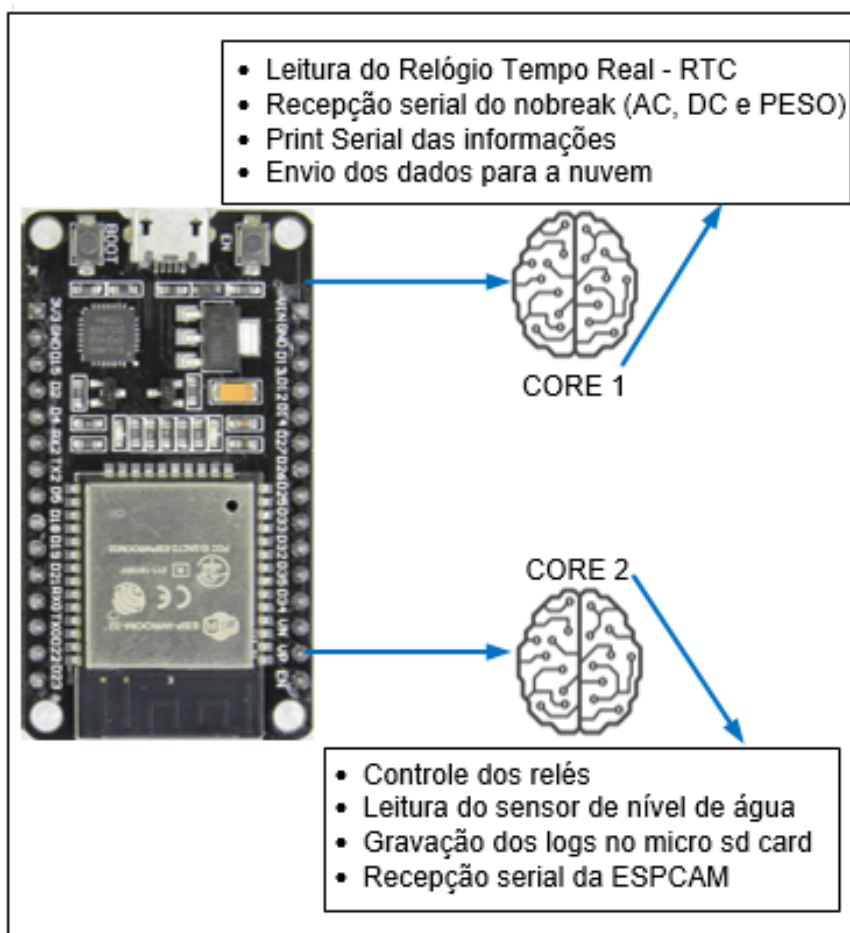
- Leitura do Relógio Tempo Real - RTC
- Recepção serial do nobreak (AC, DC e PESO)
- Print Serial das informações
- Envio dos dados para a nuvem

Já no core 2 estará rodando as funções:

- Controle dos relés
- Leitura do sensor de nível de água
- Gravação dos logs no micro sd card
- Recepção serial da ESPCAM

Esse arquivo que é gravado no *SD card*, quando quiser pode ser carregado em um programa para melhor interpretação. Quando há a necessidade de verificar os dados local do *Datalogger*? Quando o tutor estiver fora de casa e a internet parou de funcionar, desta forma, não gera registros do dispositivo na nuvem, podendo gerar dúvida se foi servido alimento ao cachorro.

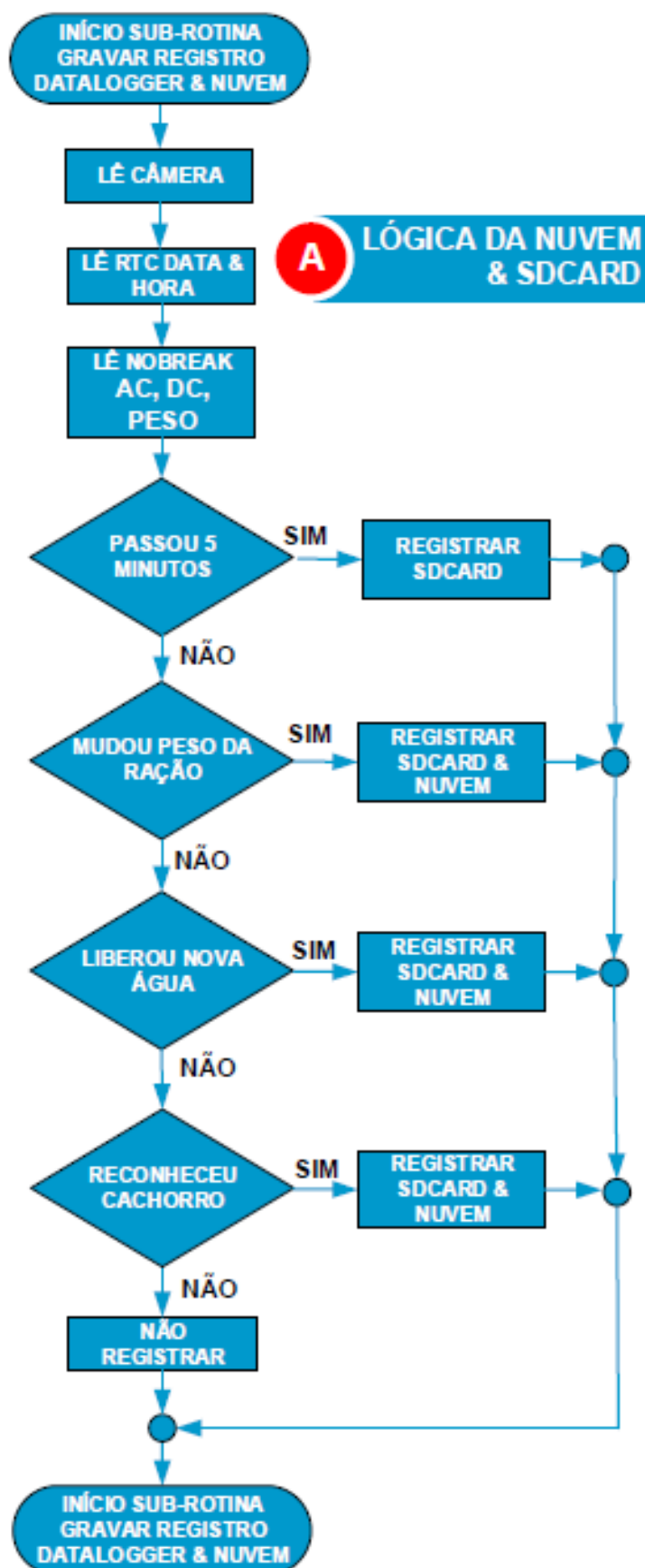
Figura 40 — lista das funções distribuídas nos dois núcleos do programa do dispositivo IoT



Fonte: Elaboração própria, 2024

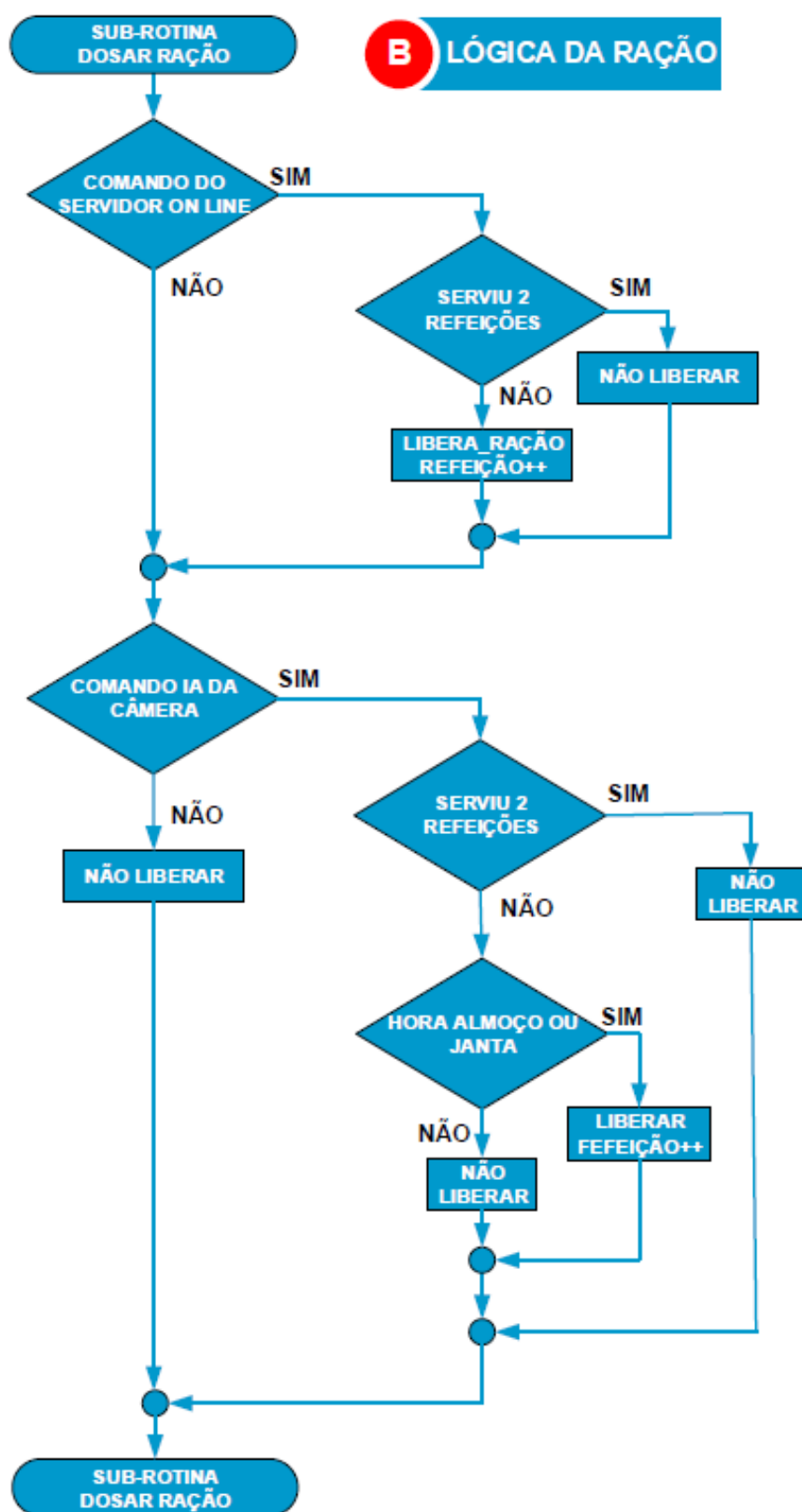
Nas Figuras a seguir, é demonstrada a lógica da programação do microcontrolador ESP32 para o sistema IoT e *Datalogger*. Devido à complexidade do sistema, a lógica está dividida em quatro partes. Na Figura 41, está representada a lógica completa da publicação na nuvem e do cartão SD. A Figura 42 detalha a lógica de controle da ração. É importante destacar que a liberação da ração pode ser realizada tanto pelo *Dashboard* do servidor *Ubidots* quanto através do reconhecimento de imagem da câmera por classificação de IA. Na Figura 43C, é demonstrada a lógica e o controle do nível da água. Por fim, na Figura 43D, encontra-se a lógica e o controle da lâmpada.

Figura 41 — Condições para registrar na nuvem e gravar no cartão SD



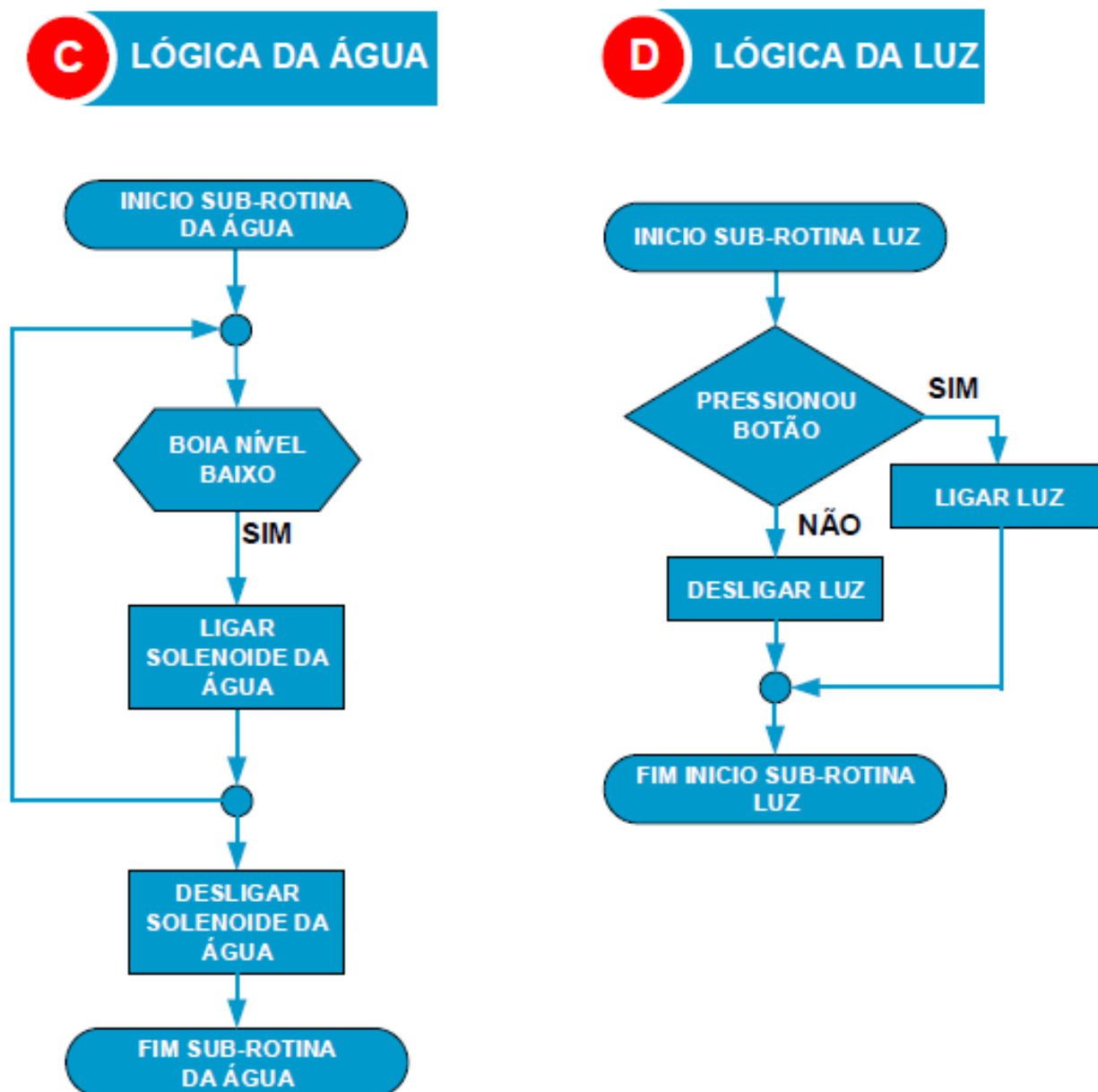
Fonte: Elaboração própria, 2024

Figura 42 — Lógica de controle da ração



Fonte: Elaboração própria, 2024

Figura 43 — C) Condição para dosar água automaticamente. D) Lógica para acender e apagar a luz do ambiente do animal de estimação.



Fonte: Elaboração própria, 2024

Para facilitar o entendimento do programa e controle do Servidor Broker, é descrito um exemplo de uma pequena parte do sistema proposto. O exemplo a seguir do código está alinhado totalmente com o exemplo dado na configuração da plataforma *Ubidots*. Aqui, é apresentada apenas uma variável de nome AD34 sendo publicada (Publish) no Servidor Broker MQTT e apenas um botão (Subscribe)

assinando no servidor Broker. Quando o botão é acionado no servidor, a mensagem é recebida na ESP32C, que por sua vez liga ou desliga o relé da GPIO27. As outras entradas e saídas são tratadas completamente no programa principal do *Datalogger*. Esse programa será apresentado nas Figuras (44, 45, 46 e 47).

Primeiramente, foi baixada uma biblioteca do GitHub no endereço: <<https://github.com/UbidoTs/esp32-mqtt/tree/main/examples>>, onde o autor deixou exemplos didáticos de publish e subscribe no servidor Broker - *UbidoTs*. Na Figura 44, está contida toda a parte de cabeçalho, inclusão da biblioteca, declaração de variáveis, configuração dos parâmetros de autenticação no servidor e conexão Wi-Fi. Na Figura 45, está demonstrada a função de callback, responsável por perceber que algo foi recebido (subscribe) do servidor Broker, indicando uma ação na ESP32. Se o valor recebido for 0 (zero), o relé da ração será desligado, e se for igual a 1, o relé da ração será ligado. Na Figura 46, tem-se a parte do Setup, que realiza a inicialização dos serviços e configuração do pino de saída. A Figura 47 é a função principal (main). Nela, a cada 5 segundos, a ESP32 reconecta-se ao servidor Broker e publica o valor AD lido no pino GPIO34 do dispositivo ESP32C no *UbidoTs*.

Figura 44 — Cabeçalho, configuração do servidor *UbidoTs* e Wi-fi

```

9  /*****
10  * Include Libraries
11  *****/
12  #include "UbidoTsEsp32Mqtt.h"
13  /*****
14  * Define Constants
15  *****/
16  const char *UBIDOTS_TOKEN = "████████████████████████████████████████"; //INSERIR SEU TOKEN DO UBIDOTS
17  const char *WIFI_SSID = "_Maximos-2g_"; // NOME DO SEU ROTEADOR WIFI
18  const char *WIFI_PASS = "lup"; //SENHA DO ROTEADOR WIFI
19  const char *PUBLISH_DEVICE_LABEL = "ESP32C"; //NOME DO DISPOSITIVO (COISA) QUE IRÁ PUBLUCAR
20  const char *PUBLISH_VARIABLE_LABEL = "AD34"; // NOME DE UMA VARIÁVEL LEITURA AD - PINO GPIO34
21  const char *SUBSCRIBE_VARIABLE_LABEL = "btracao"; // NOME DADO BOTÃO USADO NO DASHBOARD UBIDOTS
22  const int PUBLISH_FREQUENCY = 5000; // TEMPO PARA PUBLICAÇÃO (A CADA 5 SEGUNDOS PUBLICA VALOR AD)
23  unsigned long timer; // VARIÁVEL TIPO LONG
24  uint8_t analogPin = 34; // AD NO PINO GPIO34 ADC_CH6.
25  const uint8_t pinRacao = 27; GPIO27 - IO DE SAÍDA COM RELE
26  UbidoTs ubidoTs(UBIDOTS_TOKEN); // FUNÇÃO REGISTRA O SEU TOKEN DE ACESSO DO SITE

```

Fonte: Elaboração própria, 2024

Figura 45 — Função callback

```

/*****
 * Auxiliar Functions
 *****/
// FUNÇÃO CALLBACK - ESP32 FICA ESPERANDO SE UM CELULAR OU PC MANDOU MENSAGEM PARA ESP32
// SE CHEGAR 0 RELE DA RAÇÃO = DESLIGADO / SE CHEGAR 1 RELE DA RAÇÃO = LIGADO
void callback(char *topic, byte *payload, unsigned int length)
{
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++)
    {
        Serial.print((char)payload[i]);
        if ((char)payload[0] == '1')
        {
            digitalWrite(pinRacao, HIGH);
        }
        else
        {
            digitalWrite(pinRacao, LOW);
        }
    }
    Serial.println();
}

```

Fonte: Elaboração própria, 2024

Figura 46 — Função setup

```

// INICIALIZA OS SERVIÇOS / CONFIGURA SAÍDA E PREPARA O SERVIDOR UBIDOTS
void setup()
{
    // put your setup code here, to run once:
    Serial.begin(115200);
    // ubidots.setDebug(true); // uncomment this to make debug messages available
    ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
    ubidots.setCallback(callback);
    ubidots.setup();
    ubidots.reconnect();
    ubidots.subscribeLastValue(PUBLISH_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL);
    pinMode(LUZ, OUTPUT);
    timer = millis();
}

```

Fonte: Elaboração própria, 2024

Figura 47 — Função principal

```

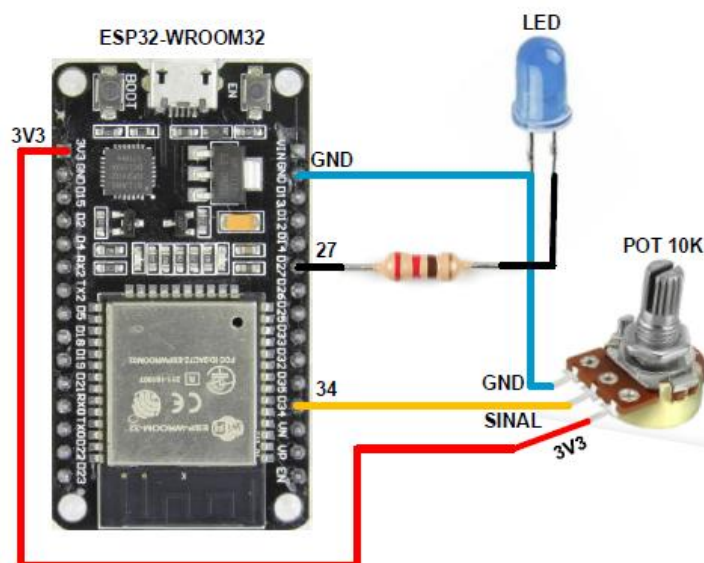
71 //*****
72 // ESTA FUNÇÃO INSERE OS PARAMETROS, CONECTA NO SERVIDOR BROKEN E
73 // PUBLICA O VALOR DA VARIÁVEL AD NO GPIO34 A CADA 5 SEGUNDOS
74 //*****
75 void loop()
76 {
77     // put your main code here, to run repeatedly:
78     if (!ubidots.connected())
79     {
80         ubidots.reconnect();
81         ubidots.subscribeLastValue(PUBLISH_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL);
82     }
83     if ((millis() - timer) > PUBLISH_FREQUENCY) // triggers the routine every 5 seconds
84     {
85         float value = analogRead(analogPin);
86         ubidots.add(PUBLISH_VARIABLE_LABEL, value); |
87         ubidots.publish(PUBLISH_DEVICE_LABEL);
88         timer = millis();
89     }
90
91     ubidots.loop();
92 }

```

Fonte: Elaboração própria, 2024

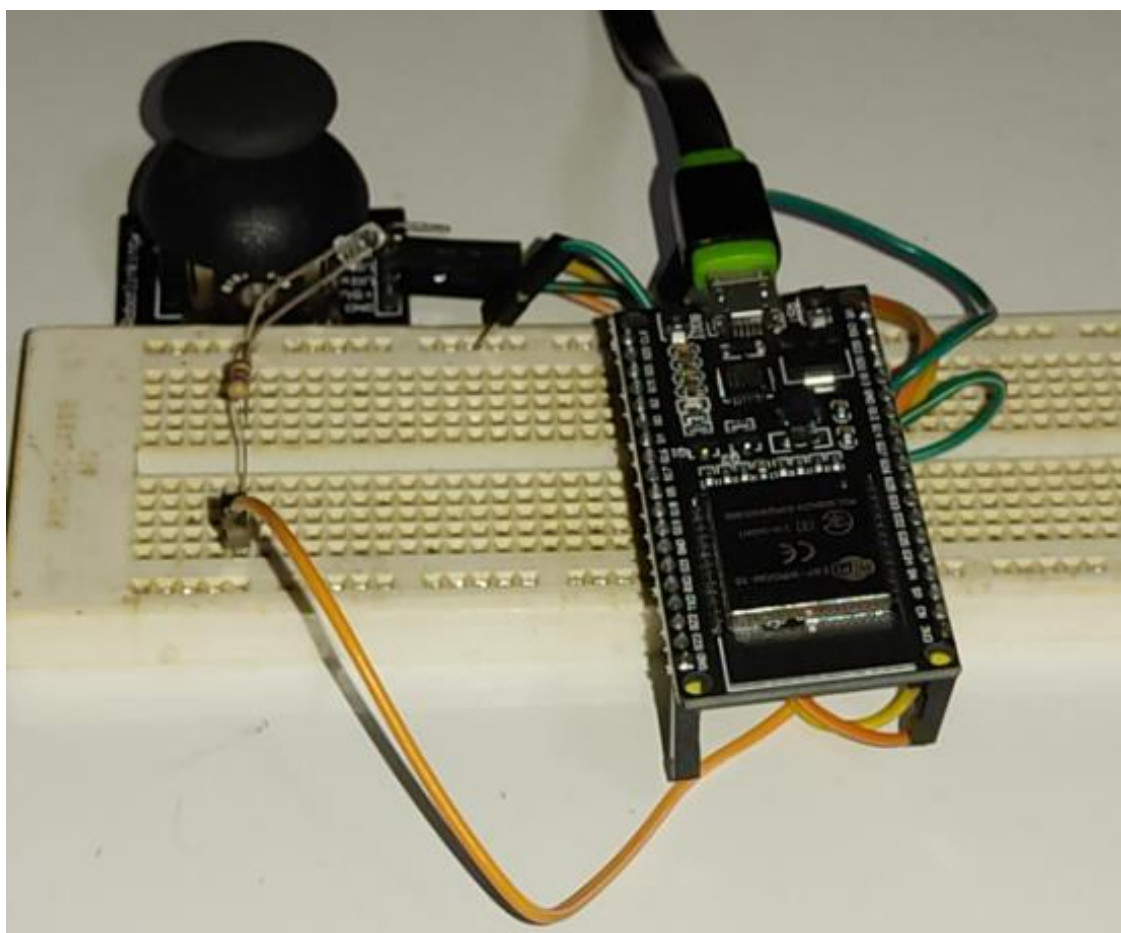
Nas Figuras 48 e 49, é possível verificar o hardware utilizado no teste e validação do programa na IDE Arduino e do servidor IoT. Esse hardware é composto por uma ESP32, um potenciômetro, um resistor de 220 ohms e um LED azul (representando o relé da ração, conectado à GPIO27), um potenciômetro como entrada e um LED como saída (conectado ao pino GPIO34).

Figura 48 — Esquema do Hardware usado da validação do programa na IDE Arduino e do servidor *Ubidots*



Fonte: Elaboração própria, 2024

Figura 49 — Hardware usado na validação do programa na IDE Arduino e do servidor *Ubidots*

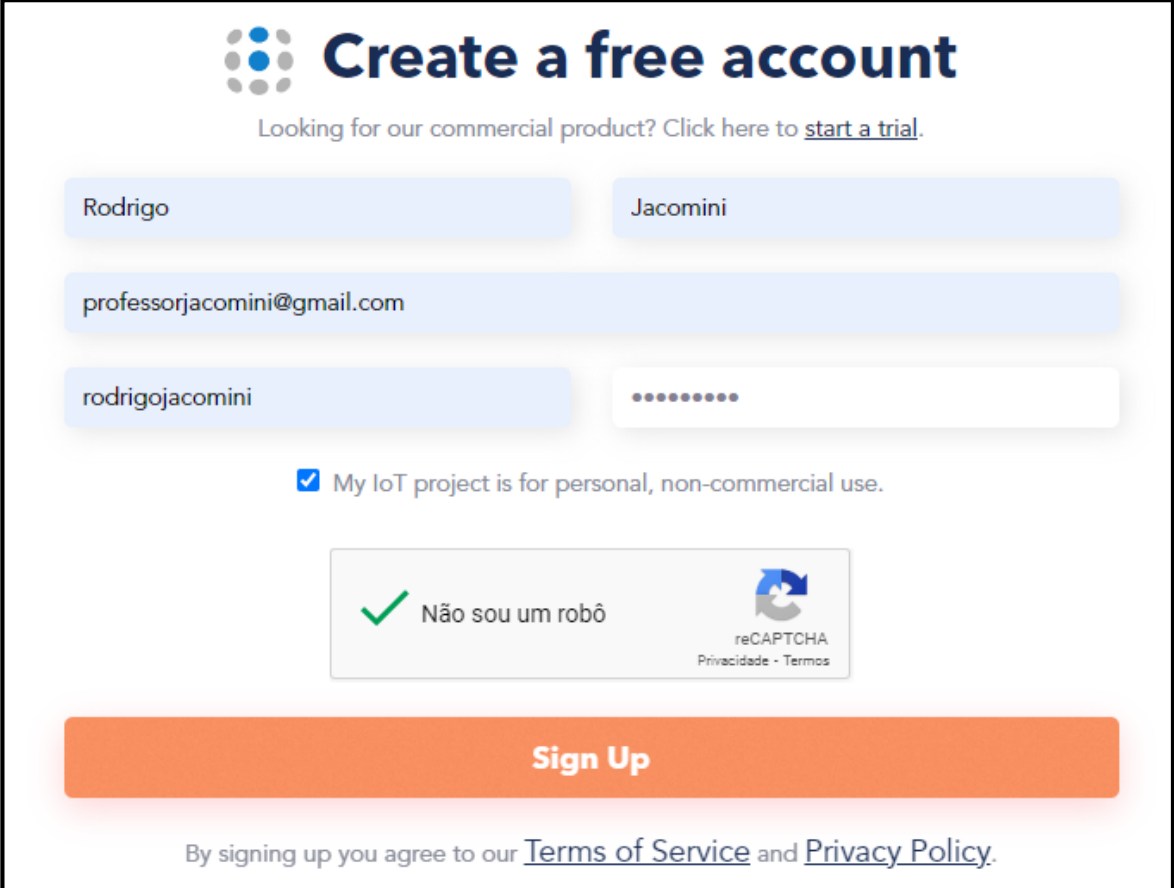


Fonte: Elaboração própria, 2024

7.9.3 IoT parte 3 – Ubidots – o Servidor

O *Ubidots* é um servidor Broker MQTT com funcionalidades semelhantes à AWS, permitindo a gestão de dispositivos pela web. Uma conta pode ser criada gratuitamente. Porém, após 30 dias, é necessário aderir a um plano para continuar utilizando a plataforma. Na Figura 50, é apresentado o processo de criação de uma conta, seguido pela explicação de como gerenciar um dispositivo, publicando no servidor Broker. Além disso, é detalhado como criar um *Dashboard* para visualização e controle do dispositivo por meio do *Ubidots* MQTT.

Figura 50 — Criação de uma conta gratuita na plataforma *Ubidots*



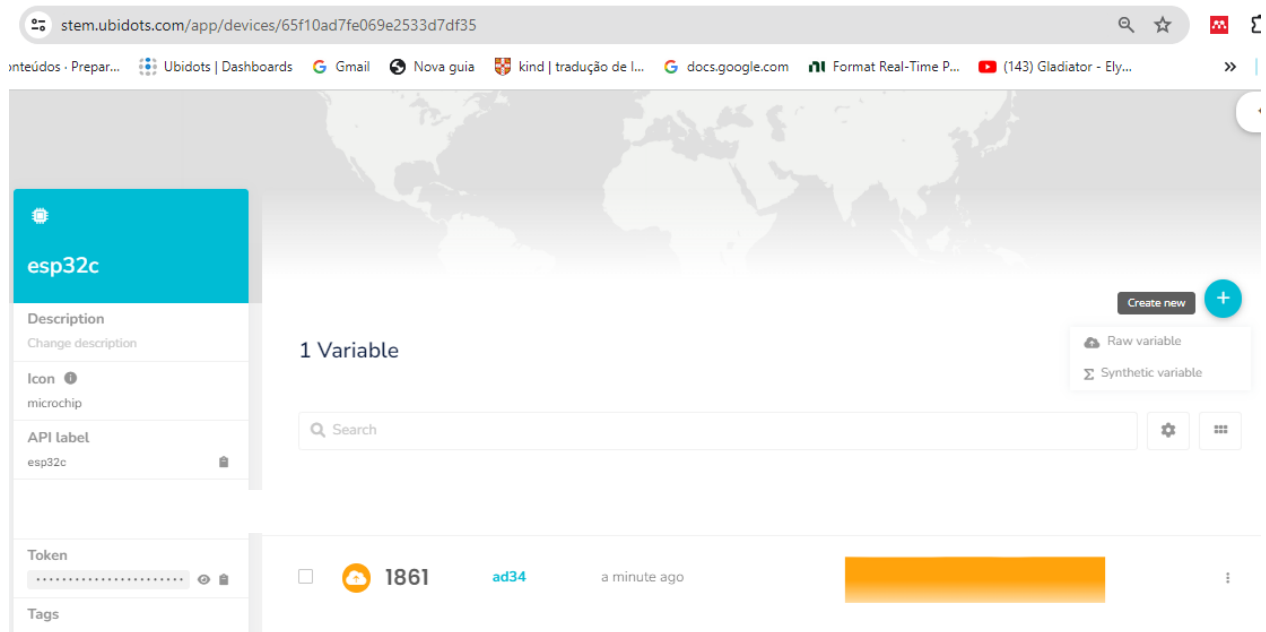
The image shows the 'Create a free account' page on the Ubidots website. At the top, there is the Ubidots logo (a cluster of blue dots) and the title 'Create a free account' in a large, bold, dark blue font. Below the title, a link says 'Looking for our commercial product? Click here to [start a trial](#).' The form contains several input fields: a first name field with 'Rodrigo', a last name field with 'Jacomini', an email field with 'professorjacomini@gmail.com', and a password field with 'rodrigojacomini'. Below these fields is a checkbox labeled 'My IoT project is for personal, non-commercial use.' which is checked. Underneath the checkbox is a reCAPTCHA verification area showing a green checkmark and the text 'Não sou um robô' (I am not a robot), along with the reCAPTCHA logo and links for 'Privacidade' (Privacy) and 'Termos' (Terms). At the bottom of the form is a large orange button labeled 'Sign Up'. Below the button, a line of text states 'By signing up you agree to our [Terms of Service](#) and [Privacy Policy](#).'

Fonte: Elaboração própria, 2024

Após gravar o código fornecido como exemplo, o microcontrolador ESP32 é reiniciado e estabelece conexão com o roteador Wi-Fi e o servidor *Ubidots*. Em seguida, inicia o processo de envio dos dados do potenciômetro para a nuvem. Ao acessar a plataforma e procurar pelo dispositivo "ESP32C", é possível visualizar os dados, como mostrado na Figura 51 abaixo. No código, é definida a variável AD34

como do tipo float, a qual é utilizada para realizar a leitura analógica do potenciômetro conectado ao pino GPIO34. Essa leitura é então enviada para o servidor *Ubidots*, permitindo o monitoramento remoto dos dados coletados.

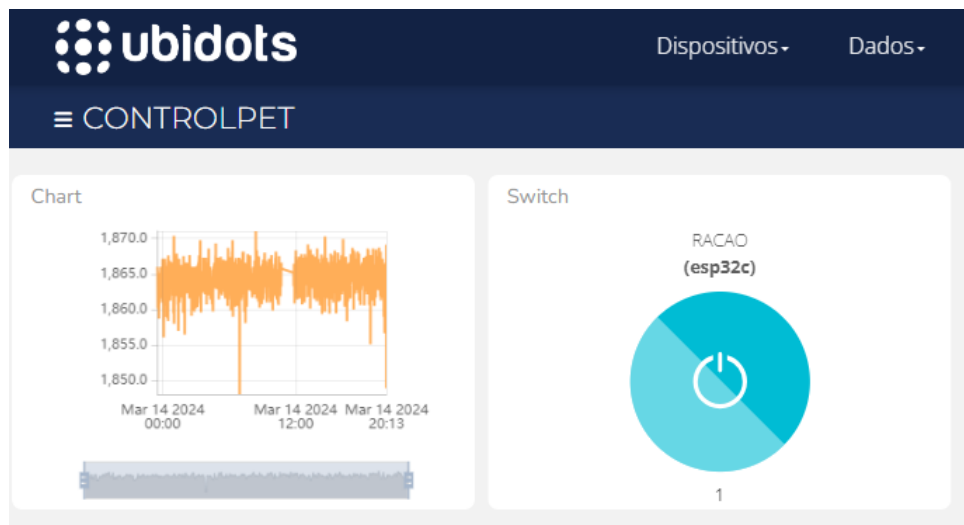
Figura 51 — Dispositivo esp32C ativo, publicando valores do potenciômetro



Fonte: Elaboração própria, 2024

Na Figura 52, é apresentada a criação da nova variável de botão de ração no *Dashboard* denominado "ControlPet". Com isso, o *Dashboard* agora exibe as duas variáveis, AD34 e o botão de ração, proporcionando uma interface completa para monitorar e controlar o sistema remotamente. Esta Figura mostra a interface via navegador em um computador para controlar a ESP32. Já as Figuras 53A e 53B demonstram a visualização e o controle das variáveis por meio de um celular. Quando o botão está cinza, ele envia o valor zero para a ESP32, desligando o LED. Ao pressionar o botão e ficar azul, é enviada a mensagem 1 para a ESP32, ligando a saída GPIO 27 do dispositivo. Essas imagens ilustram a facilidade e a versatilidade do controle remoto do sistema.

Figura 52 — Dashborad aberto no PC – Controle das Variável AD34 e Botão da Ração



Fonte: Elaboração própria, 2024

Figura 53 — *Dashboard* pronto e aberto no celular



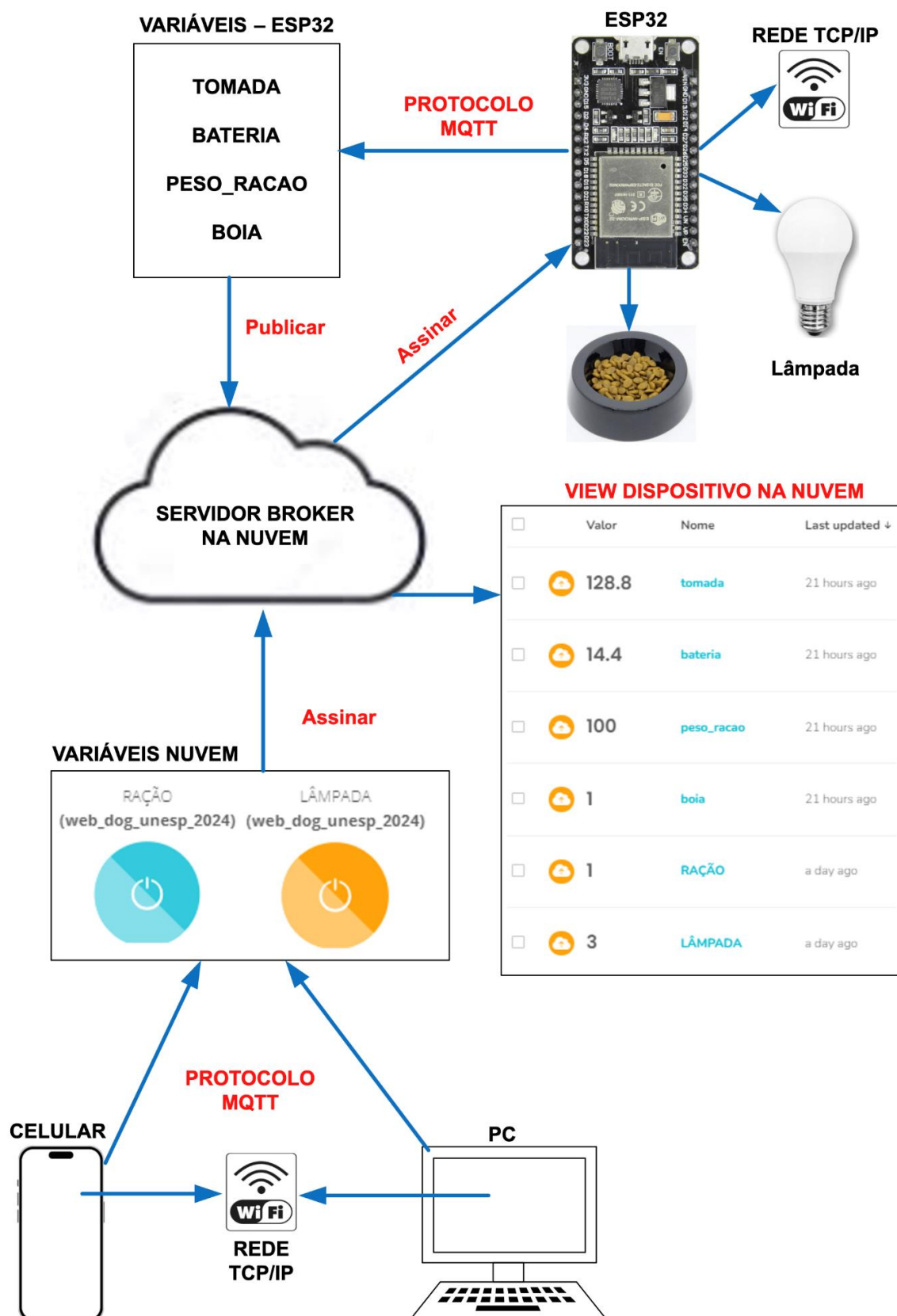
Fonte: Elaboração própria, 2024

A) Comando de LED_OFF. B) Comando de LED_ON ao pressionar o botão RACAO no celular.

7.9.4 Ubidots aplicado ao projeto da dissertação

O escopo do projeto IoT está representado como diagrama de blocos na Figura 54 abaixo. Nela é possível verificar o fluxo dos dados e como o Broker recebe os dados de publish e subscribe.

Figura 54 — Projeto IoT e fluxo dos dados publish e subscribe

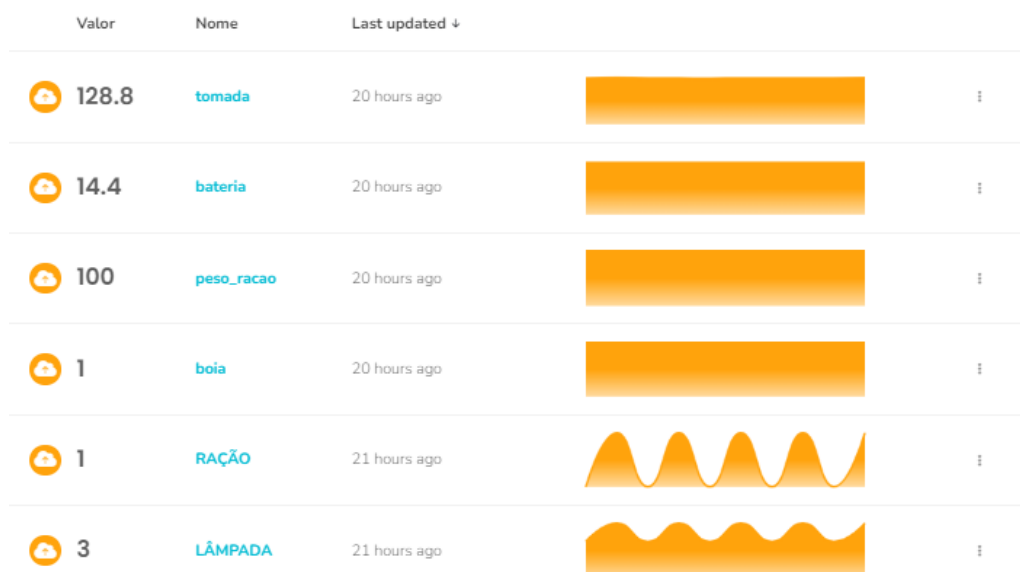


Fonte: Elaboração própria, 2024

Agora, com a implementação do novo escopo do projeto, ao acessar o dispositivo, são exibidas todas as variáveis que estão sendo enviadas e registradas no servidor Broker. Na Figura 55 abaixo, são apresentadas um total de 6 variáveis, e abaixo estão os detalhes de cada uma delas:

- **Variável tomada** (publish) serve para registrar o status da rede elétrica da tomada. Essa variável é calcula na placa do nobreak, nela contém um sensor 220 V AC;
- **A variável bateria** (publish) também é calculada no circuito do nobreak e serve para ler o valor do sinal da tensão da fonte CC de 40 A ou o valor da bateria da autonomia;
- **A variável peso_ração** (publish) serve para guardar informações a respeito da quantidade de ração no prato. Existe uma célula de carga responsável em medir o peso da ração no prato do cachorro. Essa medida serve para entender peso da ração do antes e depois das refeições. Desta forma, é capaz de avaliar a atividade de refeição do cachorro e saber a quantidade de ração dispensada e quando o cachorro se alimentou;
- **A variável boia** (publish) é um sensor magnético serve para verificar o nível água, facilitando através do *Dashboard* saber quando esvaziou, quando encheu e qual foi o consumo de água pelo cachorro;
- **A variável ração** (Subscribe), trata-se de um botão virtual adicionado no dispositivo on-line no *Ubidots* e *Dashboard*. Esse botão está associado ao *label* dado no programa da ESP32, desta forma, ao pressionar o botão no navegador ou no celular, é atualizado no dispositivo físico ESP32 que por sua vez, aciona a saída do relé da ração.
- **A variável Lâmpada** (Suscribe) também é um botão virtual no dispositivo e no *Dashboard*. Ao pressionar o botão via browser ou celular, atuará dentro da ESP32 que ligará o relé da lâmpada.

Figura 55 — Dispositivo publicando no *Ubidots* com suas 6 variáveis da pesquisa



Fonte: Elaboração própria, 2024

As Figuras 56, 57, 58, 59 e 60 apresentam as 6 variáveis abertas individualmente dentro do dispositivo (*web_dog_unesp_2024*) no servidor *Ubidots*. A Figura 56, é a variável do botão virtual da ração no *Dashboard* do servidor Broker. A Figura 57, é a variável da boia física ligada a ESP32. A Figura 58, é a variável da célula de carga ligado fisicamente no alimentador para pesar a ração servida. A Figura 59, é a variável que mede a tensão DC da bateria, esse sensor é ligado fisicamente na placa do nobreak. Esse sinal DC é enviado serialmente a placa do *Datalogger*. A Figura 60 apresenta a variável tomada, esse é um sensor físico ligado a placa do nobreak, que mede o valor de 127 V AC e também envia esse valor a placa do *datalogger*.

Abaixo, são apresentados resultados de uma simulação controlada realizada em laboratório de cada uma das variáveis.

Figura 56 — Variável botão virtual da ração



Fonte: Elaboração própria, 2024

No eixo y mostra 0 = botão despressionado e 1 = botão pressionado. No eixo x mostra data e hora do registro.

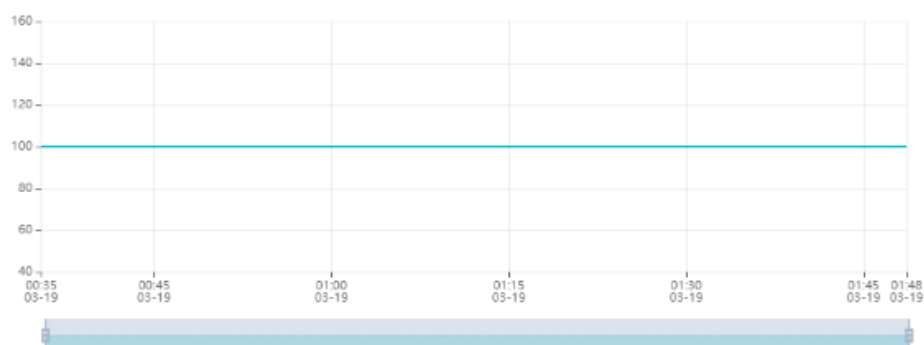
Figura 57 — Variável da boia física ligada a ESP32



Fonte: Elaboração própria, 2024

No eixo y mostra 0 = pote de água vazio e 1 = pote de água cheio. No eixo x mostra data e hora do registro

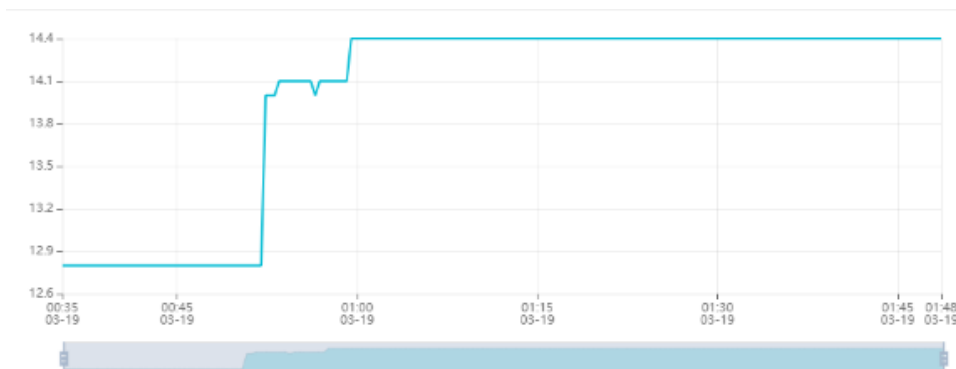
Figura 58 — Variável da célula de carga da balança



Fonte: Elaboração própria, 2024

No eixo y = demonstra quanto tem de ração em gramas. No eixo = data e hora do registro.

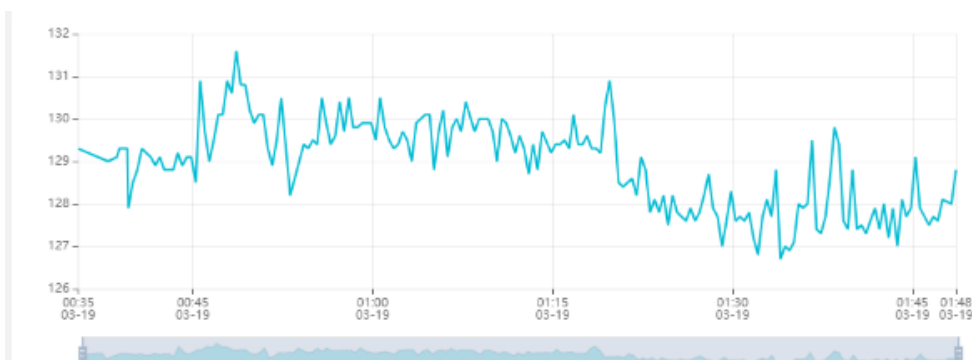
Figura 59 — Variável de medição da tensão DC da bateria



Fonte: Elaboração própria, 2024

No eixo y mostra o valor da tensão da bateria. No eixo x = data e hora do registro.

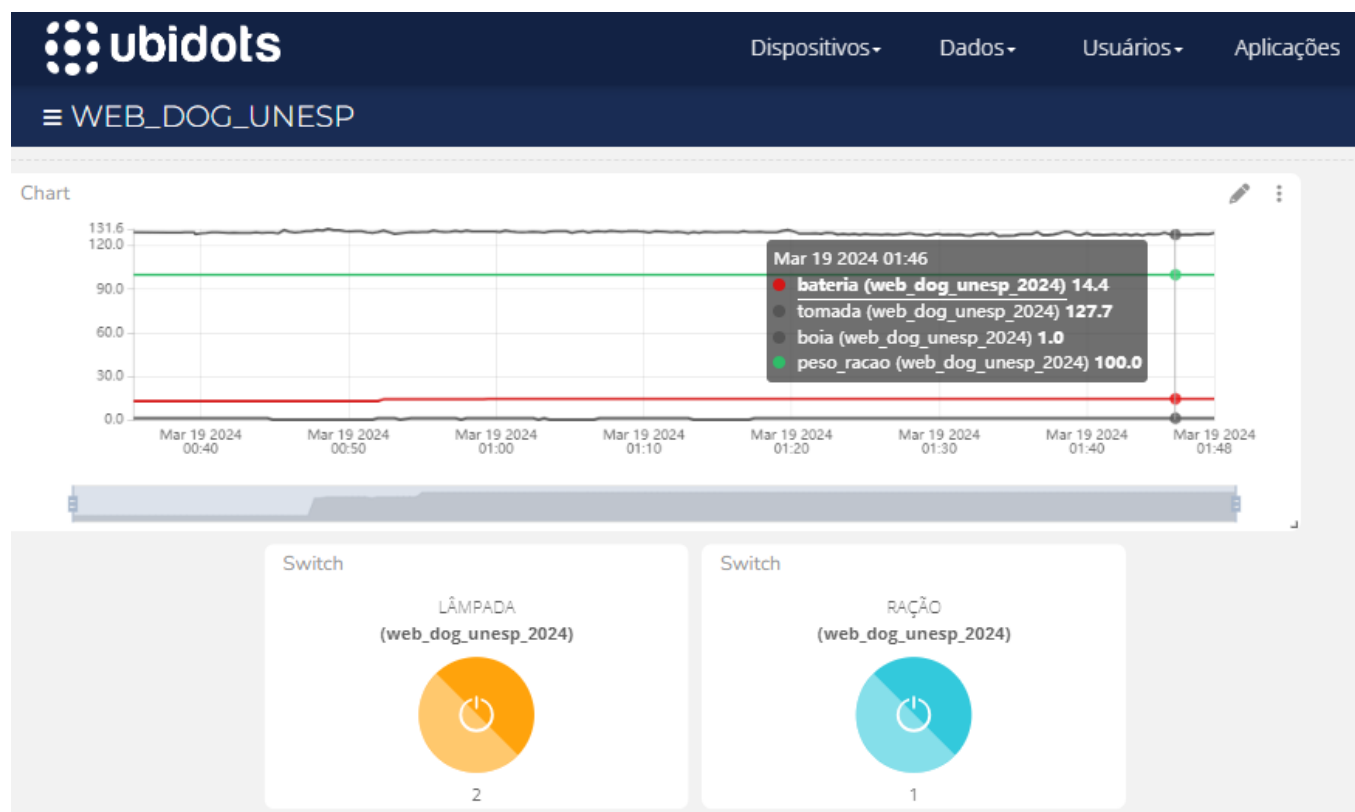
Figura 60 — Variável tomada que mede o valor de 127 V AC.



Fonte: Elaboração própria, 2024

No eixo y mostra o valor da tensão AC da tomada. No eixo x = data e hora do registro.

A seguir, na Figura 61 abaixo, é apresentado o *Dashboard* final, exibindo as variáveis de forma organizada e amigável para facilitar o entendimento no *Dashboard* "WEB_DOG_UNESP". Todas as variáveis foram agrupadas em uma única tela, proporcionando acesso simplificado em um único ponto.

Figura 61 — *Dashboard* completo do projeto da dissertação.

Fonte: Elaboração própria, 2024

8 RESULTADOS E DISCUSSÃO

Os resultados do projeto estão divididos em nove partes:

- 8.1) Resultados do Sistema de Identificação de Pet (SIP).
- 8.2) Resultado Individual do Sensor AC.
- 8.3) Resultado Individual do Sensor DC.
- 8.4) Resultados da Implementação da Célula de Carga.
- 8.5) Resultado da Autonomia da Bateria.
- 8.6) Simulação sem energia ac da rede (com nobreak operando).
- 8.7) Simulação do sistema funcionando com energia ac da rede.
- 8.8) Simulação do sistema sem Internet (*Datalogger*).
- 8.9) Teste real em casa.

8.1 RESULTADOS DO SISTEMA DE IDENTIFICAÇÃO DE PETS (SIP)

Observou-se que, para alcançar um tempo de inferência (capturar, classificar e dar resposta se era gato ou cachorro no ambiente, ou se não animal (Background)), foi necessário 1 ms. Foram utilizados no máximo 71,3 KB de memória RAM e no máximo 64,1 KB de memória Flash para o programa. Isso significa que o dispositivo de destino encarregado de executar o algoritmo de saída da rede precisa atender a esses requisitos de hardware. No caso específico da ESP32CAM, apesar de suas dimensões compactas, ela possui hardware excedente, com 4 megabytes de memória de programa Flash e 520 KB de memória RAM.

As Tabelas 2 e 3 apresentam as matrizes de confusão que avaliam a performance dos modelos YOLO e FOMO adotados. A matriz de confusão representa o desempenho do modelo de classificação em prevê as três classes: Background, Cachorro e Gato. A Diagonal principal apresenta as porcentagens de instâncias corretamente classificadas para cada classe.

O YOLO, Tabela 2, para a classe Background, obteve 99.8% das instâncias corretamente classificadas como Background; para a classe Cachorro, 83.3% das instâncias foram corretamente classificadas como Cachorro; e para a classe Gato, 66.7% das instâncias foram corretamente classificadas como Gato. Fora da diagonal principal, é possível observar as taxas de erro: 16.7% das instâncias originalmente da classe Cachorro foram erroneamente classificadas como Background, enquanto

33.3% das instâncias originalmente da classe Gato foram erroneamente classificadas como Background.

Tabela 3 — Matriz de confusão do algoritmo YOLO				
Real	Predito			
		Background	Cachorro	Gato
	Background	99,8%	0,2%	0%
	Cachorro	16,7%	83,3%	0%
	Gato	33,3%	0%	66,7%

Fonte: Elaboração própria, 2024

O FOMO, apresentado na Tabela 3, para a classe Background obteve um desempenho similar ao modelo YOLO: para a classe Cachorro, 75% das instâncias foram corretamente classificadas como Cachorro. Para a classe Gato, 50% das instâncias foram corretamente classificadas como Gato. Fora da diagonal principal, é possível observar as taxas de erro de 25% das instâncias originalmente da classe Cachorro classificadas erroneamente como Background, enquanto 50% das instâncias originalmente da classe Gato foram erroneamente classificadas como Background.

Tabela 4 — Matriz de Confusão: Treinamento do modelo aprendido usando o FOMO

Real	Predito			
		Background	Cachorro	Gato
	Background	99,8%	0,2%	0%
	Cachorro	25%	75%	0%
	Gato	50%	0%	50%

Fonte: Elaboração própria, 2024

De forma geral, tanto o modelo YOLO e FOMO na fase de treinamento foram altamente precisos em classificar instâncias da classe Background, com apenas 0.2% das instâncias sendo erroneamente classificadas como Cachorro. No entanto, ele parece ter mais dificuldade em distinguir entre as classes Cachorro e Gato, com taxas de erro consideráveis em ambas as direções.

É evidente nas Tabelas YOLO e FOMO, que mesmo usando uma abordagem

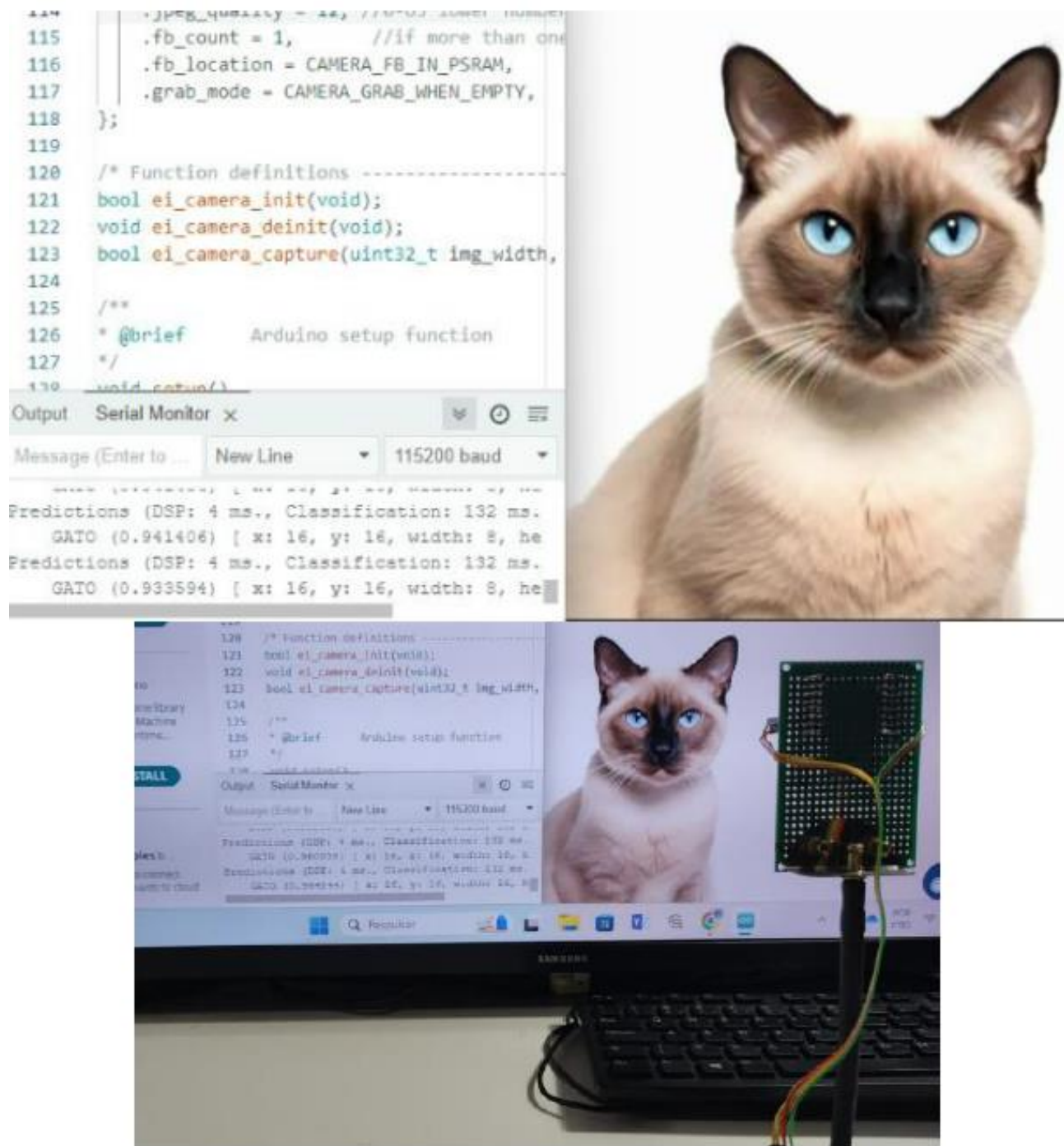
estratificada para a divisão dos dados em treinamento e teste, os modelos encontraram mais dificuldades com a classe Gato, prevendo-a corretamente apenas em 66,7% das vezes para YOLO e 50% para o FOMO. Essa matriz de confusão pode identificar onde o modelo está cometendo erros e pode orientar melhorias futuras no modelo adicionando mais imagens no treinamento para a classe Gato ou até precisar adotar uma nova abordagem como a engenharia de features para distinguir entre as características dos Gato e Background.

Na fase de teste, os modelos apresentaram, em média, uma acurácia de 75% para a classe Gato e 85% para a classe Cachorro. Ficou evidente um desequilíbrio entre as classes, com Cachorro e Gato exibindo uma distribuição mais equilibrada em comparação com a classe Background, que foi predominante. Essa disparidade pode comprometer a precisão do modelo, especialmente na ausência de estratégias de tratamento de desequilíbrio de classe durante o treinamento. Análises e ajustes futuros terão como objetivo melhorar o nível de diferenciação das classes Cachorro e Gato.

Esses resultados ressaltam as contribuições do sistema para o campo, mas também suas possíveis aplicações, enfatizando a eficácia dos modelos de classificação desenvolvidos e fornecendo informações valiosas sobre sua capacidade de distinguir entre classes diversas, como Cachorro e Gato em um sistema embarcado com recursos limitados. Além disso, o sistema também pode ser adaptado para identificar outros animais ou até mesmo pessoas, ampliando seu potencial de uso em diferentes cenários e contextos. Foi confirmado que o treinamento da rede neural foi concluído com sucesso, como evidenciado pelo excelente desempenho apresentado na matriz de confusão gerada pelo *Edge Impulse*. Outros estudos que utilizaram FOMO e YOLO para classificação de animais observaram resultados similares (Faria, Melo e Muller, 2023; Macedo, 2023). No entanto, em um estudo da área de urbanismo, foi descrito que FOMO performou melhor que YOLO na detecção de buracos na pista.

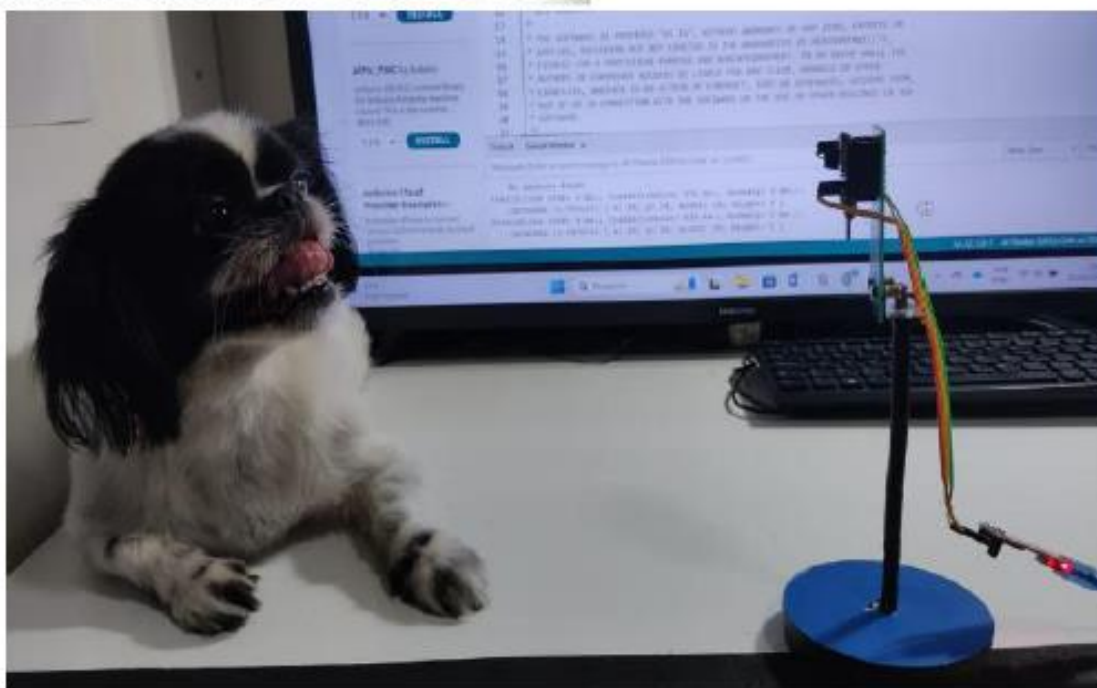
A aplicação foi adotada e, apesar de suas dimensões compactas, possui hardware excedente, com 4 megabytes de memória de programa Flash e 520 KB de RAM. As Figuras 62 e 63 demonstram o resultado da classificação da rede neural, que foi capaz de distinguir entre três grupos distintos: Gato, Cachorro e Background.

Figura 62 — Classificação (Gato) pela ESP32_CAM



Fonte: Elaboração própria, 2024

Figura 63 — Classificação (Cachorro) pela ESP32_CAM



Fonte: Elaboração própria, 2024

8.2 RESULTADO INDIVIDUAL DO SENSOR AC

O sensor de tensão AC demonstrou precisão ao medir valores na faixa de 0 a 1000 V. Os resultados dos testes, conforme ilustrado na Figura 64, mostraram que o valor exibido no display OLED correspondeu exatamente ao valor medido pelo

uma saída estável e proporcional à entrada de tensão aplicada, conforme esperado. A utilização de um divisor de tensão simples, composto por dois resistores em série, revelou-se eficaz para adaptar a faixa de tensão de entrada ao intervalo de operação do microcontrolador, garantindo uma leitura precisa e segura. Esses resultados indicam que o sensor DC é uma solução viável para aplicações que requerem medição de tensão em circuitos de baixa e média potência.

Figura 65 — Resultado do Sensor DC.



Fonte: Elaboração própria, 2024

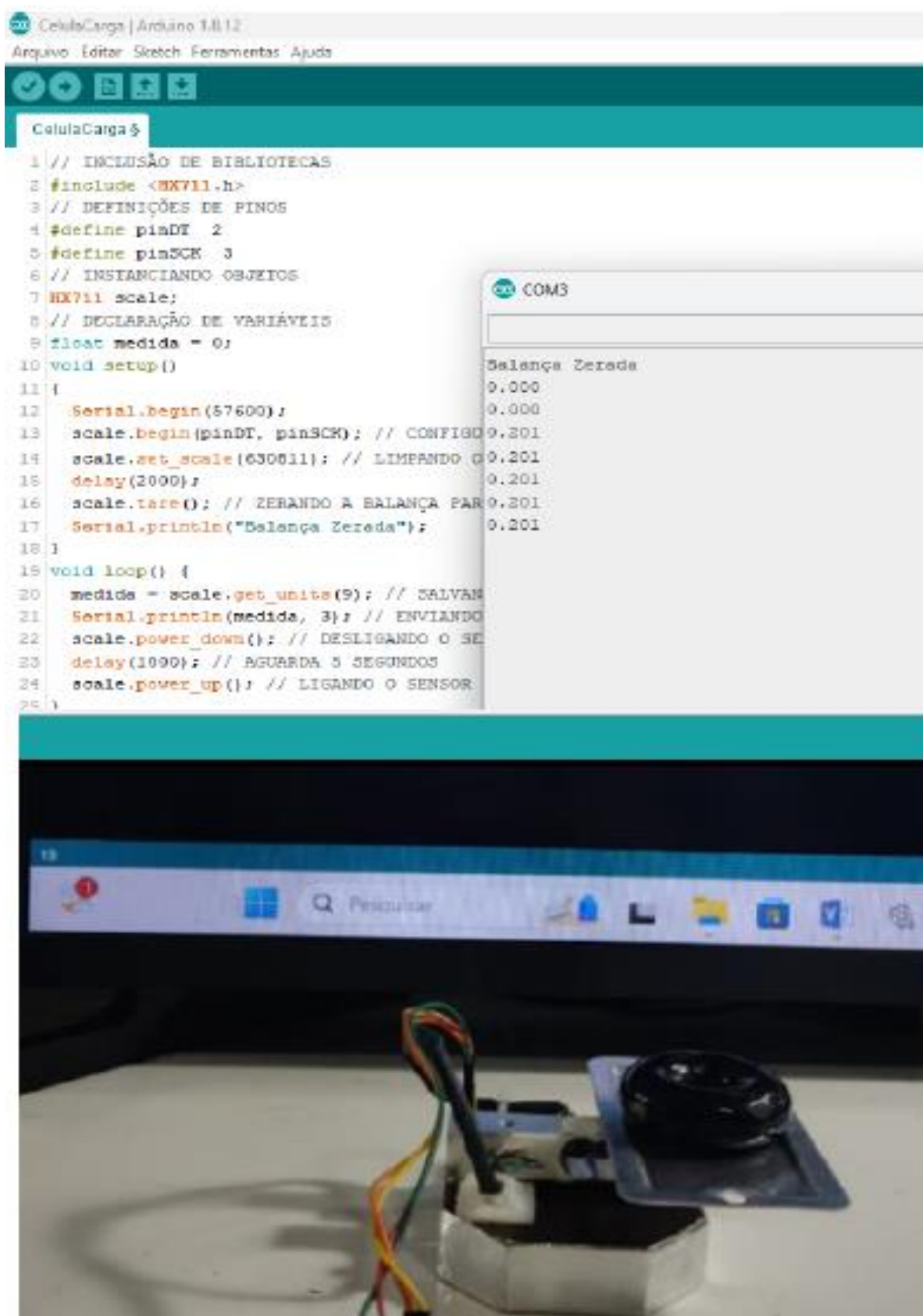
- 1) Escala do voltímetro em DC. 2) Oled do nobreak imprimindo valor de 10.9 V CC medido pelo Atmega e sensor AC. 3) Valor de medição bateria com voltímetro.

8.4 RESULTADOS DA IMPLEMENTAÇÃO DA CÉLULA DE CARGA

A implementação da célula de carga no sistema demonstrou resultados promissores em termos de medição do peso da ração no Nobreak Inteligente. Durante os testes realizados, a célula de carga foi capaz de detectar e medir com precisão o peso da ração colocada no prato da balança.

Os resultados obtidos, conforme ilustrado na Figura 66, destacam que a célula de carga integrada ao Nobreak Inteligente apresenta precisão na medição do peso da ração. Isso estabelece uma base sólida para o funcionamento autônomo do sistema de alimentação do animal de estimação.

Figura 66 — Balança funcionando, ao ligar zera a tara e depois é adicionado o peso de 0,2 kilo e começa a impressão do valor correto



8.5 RESULTADO DA AUTONOMIA DA BATERIA

O sistema de autonomia demonstrou ser mais do que adequado para as necessidades, conforme ilustrado na Figura 67. A expectativa era de que a bateria durasse pelo menos 48 horas. Antes de desconectar a fonte da tomada, verificou-se que a bateria estava completamente carregada e em estado de flutuação. O teste foi iniciado em 28/03/2024, quando a fonte foi desconectada por volta das 22 horas. Como mostrado no gráfico abaixo (Figura 67), a autonomia da bateria começou com uma leitura de 13,1 V CC. No dia 31/03/2024, a leitura ainda estava acima de 12 V CC, dentro do valor nominal da bateria. Isso indica que o sistema de autonomia está funcionando melhor do que o esperado. Em 01/04/2024, o sistema de carregamento foi reiniciado, e a bateria foi carregada utilizando um sistema de carga lenta, com a corrente controlada em até 7 A, conforme as especificações do fabricante.

Figura 67 — Resultado da autonomia de 60 horas da bateria



Fonte: Elaboração própria, 2024

8.6 SIMULAÇÃO SEM ENERGIA AC DA REDE (NOBREAK OPERANDO)

A seguir é demonstrado da Figura 68, o sistema funcionando somente na autonomia da bateria. Dados dos testes e simulação nessas condições:

- Desligamento da tomada = próximo das 22 horas do dia 28/03/2024
- Tensão da bateria no início = 13,1 V CC
- Tempo testando do sistema da Autonomia = 60 horas
- Valor da tensão no dia 31/03/2024 as 22 horas = 12,1 V CC
- Sistema religado para a recarga da bateria = dia 01/04/2024

É possível verificar que a tensão AC caiu para 0 V AC em 28/03/2024 e ficou desligada até o dia 31/03/2024. Mesmo sem AC, o sistema funcionou corretamente e foram servidas 5 refeições durante os dias da simulação (Figura 68). A água também esvaziou e se encheu automaticamente 9 vezes. Cada vez que a solenoide é acionada automaticamente, são servidos 100 ml de água, totalizando quase 1 litro em 3 dias de testes. Ao final do teste, a bateria ainda apresentava um nível ótimo de energia (12,1 V CC). Isso significa que a carga da bateria diminuiu apenas 1 V CC, o que equivale a uma diminuição de cerca de 0,016 V por hora ao longo das 60 horas de autonomia.

Esses resultados evidenciam a funcionalidade do sistema desenvolvido em comparação com outros que apenas ofereciam alimento e água para os animais de estimação, mas se mantinham sensíveis a falta de energia da rede elétrica. Atualmente, de todos os estudos descritos encontrados, nenhum descreveu o uso de uma bateria que mantivesse o sistema em funcionamento na ausência de energia elétrica.

Figura 68 — *Dashboard* do Sistema funcionando perfeitamente apenas pela bateria.

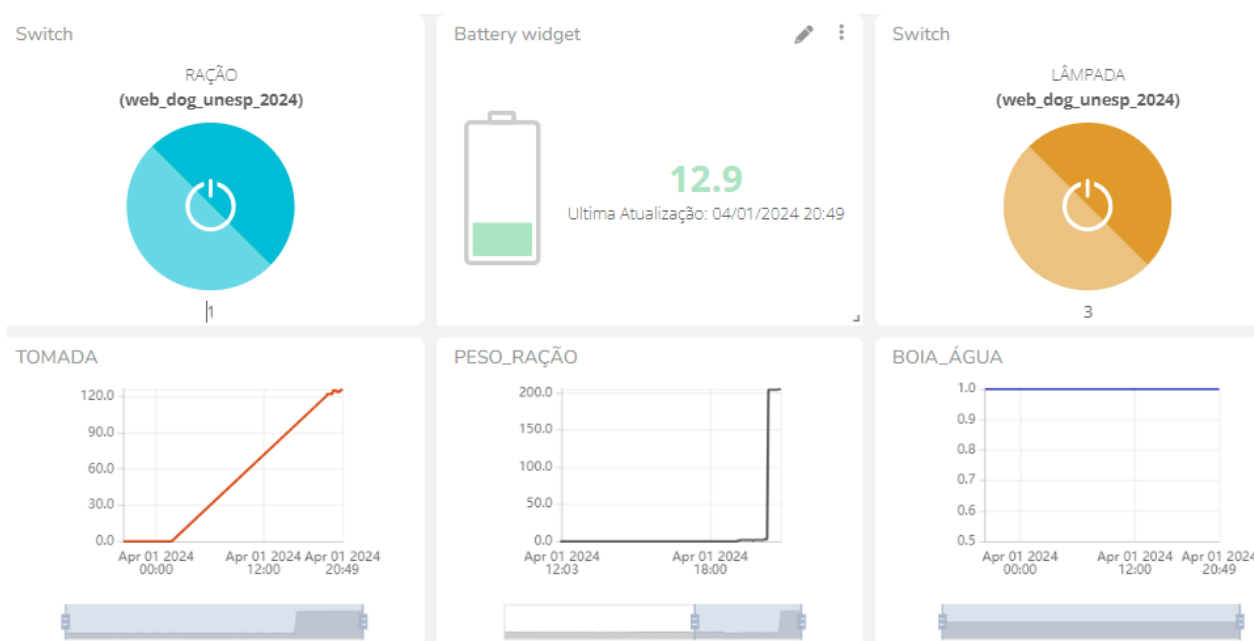


Fonte: Elaboração própria, 2024

8.7 SIMULAÇÃO COM ENERGIA AC DA REDE

No *Dashboard* apresentado na Figura 69, é retratado o que ocorreu no dia 01/04/2024, ao religar o sistema na tomada. O valor da tensão da tomada enviado para a nuvem indicou uma tensão de 124,2 V AC. A bateria estava medindo 12,9 V CC, indicando que estava em processo de carregamento. A boia estava em nível 1, o que significa que o reservatório de água estava cheio. O pote estava com 200 gramas de ração, afirmando que o sistema serviu alimento para o animal. Nesta simulação, o tutor propositalmente esqueceu de servir a ração para o animal, a fim de verificar se o sistema inteligente é capaz de auto-servir a ração usando IA. O teste de dosagem de ração foi realizado utilizando a classificação da câmera com IA. Após a cachorra Dolly ser reconhecida e classificada pela ESP32CAM, a palavra "CACHORRA" foi enviada serialmente para a ESP32 do *Datalogger*. Na ESP32, foi verificado que ainda não havia sido servida a ração na janta pelo tutor. Assim, o sistema também confirmou que estava dentro do horário programado para servir a ração e liberou uma dosagem. Na Figura 69, observa-se que o botão virtual da ração não foi acionado no dia 01/04/2024 no horário da janta. Portanto, a única forma de liberar a ração foi pela autenticação e reconhecimento do sistema de câmera.

Figura 69 — *Dashboard* do sistema AC e DC normalizados



Fonte: Elaboração própria, 2024.

8.8 SIMULAÇÃO DO SISTEMA SEM INTERNET (*DATALOGGER*)

Com a falta de energia AC da rede, o sistema continuou operando sem interrupções. No entanto, o tutor pode se preocupar se o cachorro não se alimentou, especialmente quando a informação não chega à nuvem devido à falta de internet. Para isso, além do nobreak, foi implementado um *Datalogger* para registrar eventos e atividades do cachorro. Mesmo na ausência de internet e comunicação com o servidor, o tutor terá acesso aos registros gravados localmente no cartão SD. O *Datalogger* local inclui um RTC para controlar o horário das refeições e registros. Dessa forma, a IA pode classificar e instruir a alimentação mesmo sem conectividade com a nuvem.

Outra característica importante é o controle do peso da ração, que registra mudanças no peso do pote de ração. Essas mudanças podem indicar uma nova dosagem ou que o cachorro se alimentou. Quando ocorre um evento de mudança de peso, é gerado um log instantâneo no microSD, como mostrado na Figura 70. Com isso, o tutor poderá verificar a data, a hora e a quantidade de ração e água servidas ou consumidas pelo animal na sua ausência, mesmo em casos de falta de internet e energia AC. Os campos registrados no *Datalogger* são os seguintes:

Dados existentes nos registros:

- R = Número do Registro;
- Data;
- Hora;
- Valor da Tomada;
- Peso da Ração;
- Valor da Bateria;
- Situação da Boia D'Água;
- Reconheceu o Cachorro;

Figura 70 — Arquivo interno do *Datalogger* gerado no cartão SD

Arquivo	Editar	Formatar	Exibir	Ajuda
NAME: WEB_DOG_UNESP_2024/DogLog.txt				
R;Data Hora;Tomada;PesoRacao;Bateria;SituacaoBoia;ReconheceuDog				
1;	28/3/2024	16:5:12;	211.42;	0.08;14.13;1;0
2;	28/3/2024	16:9:54;	208.09;	0.08;14.13;1;0
3;	28/3/2024	16:15:30;	210.77;	0.08;14.10;1;0
4;	28/3/2024	16:20:26;	210.84;	0.20;14.18;0;0
5;	28/3/2024	16:25:30;	211.50;	0.20;14.13;0;0
6;	28/3/2024	16:35:26;	211.83;	0.20;14.13;1;0
7;	28/3/2024	16:40:30;	211.53;	0.20;14.13;1;0
8;	28/3/2024	16:45:31;	210.34;	0.20;14.13;1;0
9;	28/3/2024	16:50:26;	212.26;	0.19;14.10;0;0
10;	28/3/2024	16:55:30;	212.48;	0.19;14.15;1;0

Fonte: Elaboração própria, 2024

Os resultados descritos acima evidenciam a importância de um *Datalogger* para que o tutor seja capaz de observar a atividade e funcionamento do sistema em sua ausência. Além disso, mesmo quando há queda de energia, o sistema se mostrou em funcionamento e sendo capaz de arquivar informações na nuvem e no cartão SD. Atualmente, na literatura, não foram encontrados estudos que utilizassem sistemas de armazenamento de informações em alimentadores de animais de estimação como desenvolvido nessa dissertação.

8.9 TESTE REAL EM CASA

Abaixo, as Figuras 71 e 72 apresentam o resultado da metodologia e o teste do protótipo com a cachorra Dolly utilizando o equipamento e se alimentando pela primeira vez em um teste real em casa. Para o treinamento da Dolly, os testes foram iniciados utilizando apenas petiscos. Isso auxiliou a cachorra a se acostumar com o equipamento como seu novo local de alimentação. O objetivo foi familiarizá-la gradualmente com o dispositivo, garantindo que ela se sentisse confortável e confiante ao usá-lo. O sistema continua em teste, e a cachorra está se acostumando com a nova invenção. Por se tratar de um treinamento, Dolly ainda está aprendendo a usar o novo sistema de identificação por captura de imagem e classificação da rede neural.

Figura 71 — Resultado físico da dissertação do Mestrado



Fonte: Elaboração própria, 2024

Figura 72 — Teste do equipamento real na casa com a cachorra



Fonte: Elaboração própria, 2024

9 CONCLUSÃO

Os resultados obtidos destacam o desenvolvimento bem-sucedido de um dispositivo inteligente que atende às necessidades dos tutores de animais de estimação, proporcionando tranquilidade e bem-estar tanto para os animais quanto para seus cuidadores. O dispositivo desenvolvido demonstrou eficiência na identificação e alimentação dos animais, além de garantir a operação contínua mesmo em situações de falta de energia elétrica. Portanto, este estudo oferece uma contribuição acadêmica significativa para o campo da tecnologia aplicada aos cuidados com animais de estimação, proporcionando soluções inovadoras e práticas para atender às necessidades dos tutores e melhorar a qualidade de vida de seus animais de estimação.

10 PERSPECTIVAS FUTURAS

No futuro, faz-se necessário adicionar mais imagens ao conjunto de treinamento e teste para as classes Gato e Cachorro como o objetivo de equilibrá-las e melhorar a capacidade dos algoritmos YOLO e FOMO. Tais ajustes trarão melhorias na distribuição das classes. Além disso, isso melhorará a capacidade do modelo de aprender características distintas de cada classe para desenvolver a um modelo mais preciso, geral e robusto na tarefa de classificação de imagens como Gatos e Cachorros e, conseqüentemente, melhorar a generalização e robustez do sistema.

Por fim, o treinamento real em casa será aperfeiçoado com a cachorra Dolly, a fim de que ela se torne independente para utilizar o alimentador. Durante o processo de treinamento, serão obtidos dados de quantas vezes a cachorra se aproxima do alimentador, se não se alimenta quando a ração é disponibilizada, entre outras informações. Esses dados serão analisados em maior profundidade futuramente.

REFERÊNCIAS

- AMAZON WEB SERVICES. **Guia do Desenvolvedor AWS IoT Core**. 2024. Washington: Amazon Web Services. Disponível em: <https://docs.aws.amazon.com/pt_br/iot/latest/developerguide/iot-dg.pdf#mqtt>. Acesso em: 18 fev. 2024.
- ARDUÍNO E CIA. **Gravador USBasp AVR com ATmega328P**. 2023. São Paulo: Arduino e Cia. Disponível em: <<https://www.arduinoecia.com.br/gravador-usbasp-avr-arduino-atmega328p/>>. Acesso em 03 fev. 2024.
- ASHTON, K. That 'Internet of Things' Thing. **RFID Journal**, 22 jun. 2009. Disponível em: <<https://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf>>. Acesso em: 12 abr. 2024.
- ASSOCIAÇÃO BRASILEIRA DA INDÚSTRIA DE PRODUTOS PARA ANIMAIS DE ESTIMAÇÃO. **Mercado Brasil Pet 2023**. São Paulo: Bela Vista, 2023. Disponível em: <https://abinpet.org.br/wp-content/uploads/2023/07/abinpet_folder_dados_mercado_2023_draft5.pdf>. Acesso em: 12 fev. 2024.
- ATMEL Corporation. **ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash Datasheet**. 2015. Califórnia: ATMEL Corporation. Disponível em: <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>. Acesso em: 29 fev. 2024.
- BEMBDE, M. et al. Automatic Pet Day-Care Robot. *In: OPJU INTERNATIONAL TECHNOLOGY CONFERENCE ON EMERGING TECHNOLOGIES FOR SUSTAINABLE DEVELOPMENT (OTCON)*, Chattisgarh, 2022. Raigarh: IEEE Xplorer, 2023, pp. 1-6. Disponível em: <<https://ieeexplore.ieee.org/document/10114041>>. Acesso em: 12 fev. 2024.
- BEMBDE, M. et al. Robotic Day-Care for Pets using Sensors and *Raspberry Pi*. *In: INTERNATIONAL CONFERENCE ON SUSTAINABLE COMPUTING AND SMART SYSTEMS (ICSCSS)*, Coimbatore, 2023. Coimbatore: IEEE Xplorer, 2023b, pp. 994–999. Disponível em: <<https://ieeexplore.ieee.org/document/10169742>>. Acesso em: 12 fev. 2024.
- BENEDITO, A. R. dos P. **Gestão técnica e comercial na indústria de pet food: a importância estratégica do Médico Veterinário neste segmento**. 2018. 34 f. Trabalho de Conclusão de Curso (Bacharelado em Medicina Veterinária), Universidade Federal Rural de Pernambuco, Recife. Disponível em: <<https://repository.ufrpe.br/handle/123456789/910>>. Acesso em: 14 jul. 2024.
- BIRHA, P. et al. Design and Development of IOT Based Pet Feeder. **International Journal of Innovations in Engineering and Science**, Nagpur, v. 7, n. 8, p. 137–140/Ago. 2022. Disponível em: <<https://doi.org/10.46335/IJIES.2022.7.8.25>>. Acesso em: 13 jul. 2024.

BOATENG, M. A.; AKPARIBO, A. R. A Multifunctional Automatic Dog-Feeder with Bluetooth and Wi-Fi Connectivity. *In: 2ND INTERNATIONAL CONFERENCE ON MOBILE NETWORKS AND WIRELESS COMMUNICATIONS (ICMNWC)*, 2, Tumkur, 2022. Karnataka: IEEE Xplorer, 2023, pp. 1-6. Disponível em: <<https://ieeexplore.ieee.org/document/10031807>>. Acesso em 15 jul. 2024.

BRANDÃO, R. **AI-Thinker ESP32-CAM Wi-Fi+BT SoC Module V1.0 ESP32-CAM Module Overview**. In: GitHub. [São Francisco, CA: GitHub Inc., 2008]. Disponível em: <https://github.com/raphaelbs/esp32-cam-ai-thinker/blob/master/assets/ESP32-CAM_Product_Specification.pdf>. Acesso em: 1 mar. 2024.

CHEN, Y.; ELSHAKANKIRI, M. Implementation of an IoT based Pet Care System. *In: INTERNATIONAL CONFERENCE ON FOG AND MOBILE EDGE COMPUTING (FMEC)*, 5, Paris, 2020. Paris: IEEE Xplorer, 2020, pp. 256-62. Disponível em: <<https://ieeexplore.ieee.org/document/9144910>>. Acesso em: 15 jul. 2024.

DALMAS, E. G. **O comportamento do consumidor de produtos e serviços do mercado pet quanto aos cuidados com os animais de estimação**. 2019. 75 f. Trabalho de Conclusão de Curso (Bacharelado em Administração), Universidade de Caixas do Sul, Rio Grande do Sul. Disponível em: <<https://repositorio.ucs.br/xmlui/handle/11338/6117>>. Acesso em: 14 jul. 2024.

DEVAKI, M. et al. Smart Energy Meter for Automatic Billing Using IoT with AWS. *In: INTERNATIONAL CONFERENCE ON VISION TOWARDS EMERGING TRENDS IN COMMUNICATION AND NETWORKING TECHNOLOGIES (VITECON)*, 2, 2023. IEEE Xplorer: 2023, pp. 1-5. Disponível em: <https://www.researchgate.net/profile/Devaki_Mohananarangam/publication/371888221_Smart_Energy_Meter_for_Automatic_Billing_Using_IoT_with_AWS/links/64c74ce4545060019e42e019/Smart-Energy-Meter-for-Automatic-Billing-Using-IoT-with-AWS.pdf>. Acesso em: 18 fev. 2024.

ELIZEIRE, M. B. **Expansão do mercado pet e a importância do marketing na medicina veterinária**. 2013. 51 f. Trabalho de Conclusão de Curso (Bacharelado em Medicina Veterinária), Universidade Federal do Rio Grande do Sul, Rio Grande do Sul. Disponível em: <<https://lume.ufrgs.br/handle/10183/80759>>. Acesso em: 14 jul. 2024.

ESPRESSIF. **Datasheet ESP32WROOM32**. Espressif Systems, 2023. Disponível em: <<https://www.espressif.com/en/support/download/documents>>. Acesso em: 10 mar. 2024.

FARIA, L.; MELO, M.; MULLER, E. Utilização de classificação de imagens em *TinyML* para auxílio na alimentação de aves. *In: CIÊNCIA COMO FERRAMENTA DE TRANSFORMAÇÃO DA SOCIEDADE*, 6, Itabira, 2023. Anais do VI Simpósio de Iniciação Científica. Itajubá: 2023. Disponível em: <<https://periodicos.unifei.edu.br/index.php/rtic/article/view/598>>. Acesso em: 13 jul. 2024.

FERCAR. **Comedouros por Gravidade**. Metalúrgica Fercar, 2024. Disponível em: <<https://fercar.ind.br/produtos/comedouros>>. Acesso em: 14 abr. 2024.

GADELHA, T. M. C. **Sistema para Monitoramento do Nível de Água em Recipientes de Animais Domésticos**. 2020. 11f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação), Universidade Federal de Campina Grande, Paraíba, Disponível em: <<http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/20341>>. Acesso em: 15 jul. 2024.

HARAHAP, R. K. et al. Dogs Feed Smart System with Food Scales Indicator IoT Based. In: **INTERNATIONAL CONFERENCE ON CYBERNETICS AND INTELLIGENT SYSTEM (ICORIS)**, 4, Prapat, 2022. Prapat: IEEE Xplorer, 2023, pp. 1-7. Disponível em: <<https://ieeexplore.ieee.org/document/10031344>>. Acesso em: 15 fev. 2024.

HAREENDRAN, T. K. **ZMPT101B AC Voltage Sensor Module Primer**. Eletro Schematics, 2022. Disponível em: <<https://www.electroschematics.com/voltage-sensor/>>. Acesso em: 13 abr. 2024.

HYMEL, S et al. *Edge Impulse: an MLOPS platform for tiny machine learning*. **ArXiv**, s.l, s.n, pp. 1-15, Abril, 2023. Disponível em: <<https://arxiv.org/pdf/2212.03332v2/>>. Acesso em: 20 fev. 2024.

HUDSON, G. **Library Emonlib.h Open Energy Monitor**. In: GitHub. [São Francisco, CA: GitHub Inc., 2008]. Disponível em: <<https://github.com/openenergymonitor/docs/blob/main/main/electricity-monitoring/ac-power-theory/advanced-maths.md>>. Acesso em: 13 abr. 2024.

JACOMINI, R. **Web Pet – Monitoramento e controle de água, ração e luz via web**. 2013. 69f. Trabalho de Conclusão de Curso (Especialização em Tecnologias e Sistemas da Informação), Universidade Federal do ABC, São Paulo.

JAIN, E. et al. H. Design of Smart Pet Food Dispenser using Embedded System. In: **INTERNATIONAL CONFERENCE ON EMERGING SMART COMPUTING AND INFORMATICS (ESCI)**, Pune, 2023. Pune: IEEE Xplorer, 2023, pp. 1-5. Disponível em: <<https://ieeexplore.ieee.org/document/10100166>>. Acesso em: 15 fev. 2024.

LIMA, M. G. et al. **Coleira rastreadora para pets**. 2023. 31f. Trabalho de Conclusão de Curso (Curso Técnico Mecatrônica Integrado ao Ensino Médio), Escola Técnica Estadual Sylvio de Mattos Carvalho, São Paulo. Disponível em: <<https://ric.cps.sp.gov.br/handle/123456789/16942>>. Acesso em: 5 abr. 2024.

MACEDO, R. **Classificação de raças de cachorro utilizando técnicas de transfer learning na plataforma Edge Impulse**. 2023. 41f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação), Universidade Federal do Rio Grande do Norte, Rio Grande do Norte. Disponível em: <https://repositorio.ufrn.br/bitstream/123456789/53192/1/TCC_RUBENS_MACEDO_PEREIRA.pdf>. Acesso em: 19 fev. 2024.

MCCULLOCH, W.; PITTS, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. **Bulletin of Mathematical Biophysics**, Oxford, v. 52, n. ½, pp. 99-115, 1990. Disponível em:
<<https://www.cs.cmu.edu/~.epxing/Class/10715/reading/McCulloch.and.Pitts.pdf>>. Acesso em 21 fev. 2024.

MENEZES, C. **Detecção de Objetos Através de Rede Neural do tipo FOMO**. In: GitHub. [São Francisco, CA: GitHub Inc., 2008]. Disponível em:
<https://github.com/cezmen/voltamo/blob/main/voltamo_proposta_projeto_01.pdf>. Acesso em: 2 mar. 2024.

MOLINA, G. Internet das Coisas (IoT) na Automação Residencial. In: **SEVEN INTERNATIONAL ENGINEERING CONGRESS**, 1, Itupeva, 2024. Itupeva: Anais SEV7N Publicações Acadêmicas, 2024. Disponível em:
<<https://sevenpublicacoes.com.br/index.php/anais7/article/view/4295>>. Acesso em: 21 jul. 2024.

NASCIMENTO, W. **Inteligência Artificial em Sistemas Embarcados Utilizando Abordagem TinyML**. 2022. 67f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação). Universidade Tecnológica Federal do Paraná, Paraná. Disponível em:
<<https://repositorio.utfpr.edu.br/jspui/bitstream/1/30638/1/inteligenciaartificialembarcaadostinyML.pdf>>. Acesso em: 20 fev. 2024.

OLIVEIRA, V. L. C. et al. Estudo dos Agronegócios 4.0 – Tecnologias, desafios e benefícios nos Agronegócios. **Research, Society and Development**, Vargem Grande Paulista, v. 11, n. 13, pp. 1-13, Outubro, 2022. Disponível em
<<https://rsdjournal.org/index.php/rsd/article/download/35379/29805/393661>>. Acesso em 18 fev. 2024.

PALOMINO, J. M. et al. Expansão da cana-de-açúcar e o impacto sobre a arrecadação fiscal dos municípios paulistas. In: **CONGRESSO DA SOCIEDADE BRASILEIRA DE ECONOMIA, ADMINISTRAÇÃO E SOCIOLOGIA RURAL**, 46., 2008, Rio Branco. Anais [...]. Rio Branco: Sober, 2008. Disponível em:
<http://www.sober.org.br/palestra/9/445.pdf>. Acesso em: 15 jun. 2012

PEREIRA, M. C. et al. Inovação na alimentação de animais de estimação: Desenvolvimento de um sistema automatizado para pets. **Revista Eletrônica de Engenharia Elétrica e Engenharia Mecânica**, Mossoró, v. 5, n. 2, pp. 43–54, Dezembro, 2023. Disponível em:
<<https://periodicos.ufersa.edu.br/r4em/article/view/12229>>. Acesso em: 14 abr. 2024.

RAMOS, T. P.; JUNIOR, V. B. **Aplicação do conceito de internet das coisas no desenvolvimento de um alimentador para animais domésticos**. 2023. 18f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação), Universidade do Extremo Sul Catarinense, Santa Catarina. Disponível em:
<<http://repositorio.unesc.net/handle/1/10356>>. Acesso em: 13 jul. 2024.

PIRES, F. et al. Alimentador Automático de Animais Domésticos Utilizando Controle de Motor de Passo. *In: CONGRESSO BRASILEIRO DE AUTOMÁTICA*, 24, Fortaleza. Sociedade Brasileira de Automática. Fortaleza: Fábrica de Negócios, 2022. Disponível em:

<https://sba.org.br/open_journal_systems/index.php/cba/article/view/3651/3180>. Acesso em: 30 set. 2024.

PRASAD, D. D. et al. Feeding Pets using Automation Technology and Google Assistant. *In: INTERNATIONAL CONFERENCE ON ELECTRONICS, COMMUNICATION AND AEROSPACE TECHNOLOGY (ICECA)*, 7, Coimbatore, 2023. IEEE Xplorer: Coimbatore, 2023, pp. 224-229. Disponível em:

<<https://ieeexplore.ieee.org/document/10395544>>. Acesso: 14 abr. 2024.

QUINTINO, L. F. et al. Kit didático de baixo custo para práticas interdisciplinares em cursos de Engenharia Eletrônica. **Revista Principia - Divulgação Científica e Tecnológica do IFPB**, João Pessoa, v. 1, n. 34, pp. 1-8, Junho, 2017. Disponível: <<https://periodicos.ifpb.edu.br/index.php/principia/article/viewFile/1341/659>>. Acesso em: 14 abr. 2024.

RIBEIRO, F. **Internet das Coisas: da Teoria à Prática**. 2019. 58f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação), Universidade Federal do Ouro Preto, Minas Gerais. Disponível em:

<https://monografias.ufop.br/bitstream/35400000/1794/1/MONOGRAFIA_InternetCoisasTeoria.pdf>. Acesso em: 14 fev. 2024.

RODRIGUES, C. F. A et al. Projeto e Desenvolvimento de um Sistema Computacional para controle e acompanhamento de alimentação diária de PETs. *In: ENCONTRO NACIONAL DE COMPUTAÇÃO DOS INSTITUTOS FEDERAIS (ENCOMPIF)*, 11, Brasília, 2024. Anais do XI Encontro Nacional de Computação dos Institutos Federais. Porto Alegre: Sociedade Brasileira de Computação, 2024. Disponível em: <<https://sol.sbc.org.br/index.php/encompif/article/view/29283>>. Acesso em: 13 jul. 2024.

RODRIGUES, J.; QUEIRÓZ, I. **Projeto IoT aplicado à construção de um alimentador automático para animais domésticos**. 2019. 12f. Trabalho de Conclusão de Curso (Bacharelado em Ciência e Tecnologia), Universidade Federal Rural do Semiárido, Rio Grande do Norte. Disponível em: <<https://repositorio.ufersa.edu.br/server/api/core/bitstreams/d0bdb2fa-73f0-40a1-b849-3f4f87a331ba/content>>. Acesso em: 14 jul. 2024.

ROSSI, D.; JOSELINO, B.; JUNIOR, X. **Protótipo de alimentador automático para animais domésticos – Gingapets**. 2017. 86f. Trabalho de Conclusão de Curso (Tecnólogo em Mecatrônica Industrial), Instituto Federal de Santa Catarina, Santa Catarina. Disponível em: <<https://joinville.ifsc.edu.br/~bibliotecajoi/arquivos/tcc/mecind/180273.pdf>>. Acesso em: 15 fev. 2024.

SAMUEL, A. L. Some Studies in Machine Learning Using the Game of Checkers. **IBM Journal of Research and Development**, s.l., v. 3, n. 3, p. 210–229, jul. 1959. Disponível em: <<http://ieeexplore.ieee.org/document/5392560/>>. Acesso em: 12 abr. 2024.

SANTOS, B. et al. Internet das Coisas: da Teoria à Prática. In: SIQUEIRA, F. A. et al. (org.) **Livro de Minicursos SBRC 2016**. 34ed. Porto Alegre, RS: Sociedade Brasileira de Computação (SBC), 2016. Capítulo 1, pp. 1-50. Disponível em: <<https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>>. Acesso em: 12 fev. 2024.

SILVA, V. H. do N. **Protótipo: alimentador inteligente para cães e gatos baseado no conceito de Internet das Coisas**. 2021. 21f. Trabalho de Conclusão de Curso (Tecnólogo(a) em Sistemas para Internet), Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, Pernambuco. Disponível em: <<http://releia.ifsertao-pe.edu.br:8080/jspui/handle/123456789/682>>. Acesso em: 13 jul. 2024.

SOARES, G. M.; TELHADO, J.; PAIXÃO, R. L. Avaliação da percepção de proprietários de cães residentes em apartamentos no município de Niterói-RJ sobre os sinais da síndrome de ansiedade de separação em animais. **Archives of Veterinary Science**, Curitiba, v. 17, n. 2, pp. 10-17, Julho, 2012. Disponível em: <https://www.researchgate.net/publication/277733612_Avaliacao_da_percepcao_de_proprietarios_de_caes_residentes_em_apartamentos_no_municipio_de_Niteroi-RJ_sobre_os_sinais_da_sindrome_de_ansiedade_de_separacao_em_animais>. Acesso em: 12 fev. 2024.

TEIXEIRA, G. J.; RODOWANSKI, I. J. Protótipo de um alimentador para cães com identificador por radiofrequência. **Revista Eletrônica de Ciências Exatas e Tecnológicas**, Cruz das Almas, v. 2, n. 1, pp. 1-12, Julho, 2021. Disponível em: <<https://www3.ufrb.edu.br/index.php/recet/article/view/2491>>. Acesso em: 15 fev. 2024.

‘TRYSTANLEA’. **AC Power Theory - Advanced Maths**. In: GitHub. [São Francisco, CA: GitHub Inc., 2008]. Disponível em: <<https://github.com/openenergymonitor/docs/blob/main/main/electricity-monitoring/ac-power-theory/advanced-maths.md>>. Acesso em: 13 abr. 2024.

VALERI, J. **Cresce o número de famílias que preferem ter pets em vez de filhos**. **Jornal da USP, São Paulo, 21 jun. 2023**. Disponível em: <<https://jornal.usp.br/campus-ribeirao-preto/cresce-o-numero-de-familias-que-preferem-ter-pets-em-vez-de-filhos/>>. Acesso em: 5 abr. 2024.

VENKATASWAMY, R.; JANAMALA, V.; CHERUKURI, R. C. Realization of Humanoid Doctor and Real-Time Diagnostics of Disease Using Internet of Things, Edge Impulse Platform, and ChatGPT Annals of Biomedical Engineering. In: **NATIONAL LIBRARY OF MEDICINE**, 52, Bethesda, 2024. Anais de Engenharia Biomédica. Bethesda: National Library of Medicine, 2024. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/37453975/>>. Acesso em: 21 fev. 2024.

VIMOS, N. et al. **Diseño e implementación de sistema de comunicación inalámbrico para monitoreo del sistema de emergencia eléctrico en nodos críticos de comunicación.** 2022. Trabalho de Conclusão do Curso (Bacharelado em Engenharia Eletrônica e Telecomunicações). Universidade Nacional de Chimborazo, Riobamba, 2022. Disponível em: <<http://dspace.unach.edu.ec/handle/51000/8858>>. Acesso em: 22 jul. 2024.

WANG, J. et al. Design of a Pet Feeder Based on Single Chip Microcomputer. *In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND AUTOMATION CONTROL (AIAC)*, Xiamen, 2023. Xiamen: IEEE Computer Society, 2024, pp. 330-333. Disponível em: <<https://ieeexplore.ieee.org/document/10491627>>. Acesso em: 15 abr. 2024.