UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

Thiago Sousa Borges da Silva

Karen Cristina Kai

Ricardo de Jesus Santos

Guilherme Felipe Ribeiro

Giovanna Maia Olivieri

Daniel Tobias Miguel

Dayane Trevisan de Macedo Almeida

Gabriela Vitoria Marguti Gil Ferreira

HYDRAPET VISION 2.0: Projeto de um Sistema Embarcado para Monitoramento e Registro Fotográfico de Visitas ao Bebedouro Felino

Vídeo de apresentação do Projeto Integrador

< https://youtu.be/FG6PT9iddwA>

São Paulo - SP
2026

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

HYDRAPET VISION 2.0: Projeto de um Sistema Embarcado para Monitoramento e Registro Fotográfico de Visitas ao Bebedouro Felino

Relatório Técnico-Científico apresentado na disciplina de Projeto Integrador VI para o curso de Engenharia de Computação da Universidade Virtual do Estado de São Paulo (UNIVESP).

São Paulo - SP
2026

DA SILVA, Thiago Sousa Borges; KAI, Karen Cristina; SANTOS, Ricardo de Jesus; RIBEIRO, Guilherme Felipe; OLIVIERI, Giovanna Maia; MIGUEL, Daniel Tobias; ALMEIDA, Dayane Trevisan de Macedo; FERREIRA, Gabriela Vitoria Marguti Gil. **HYDRAPET VISION 2.0: Projeto de um Sistema Embarcado para Monitoramento e Registro Fotográfico de Visitas ao Bebedouro Felino**. 146 p. Relatório Técnico-Científico. Bacharelado em Engenharia de Computação – **Universidade Virtual do Estado de São Paulo (UNIVESP)**. Tutor: Prof. Avelar Rodrigues de Sousa. Polos: Inácio Monteiro, Caminho do Mar, Parque Bristol, Butantã, Azul da Cor do Setembro, 2026.

RESUMO

Este trabalho apresenta o projeto, o desenvolvimento e a verificação analítica (análise estática do código, cálculos de engenharia e simulação conceitual) da versão 2.0 do HydraPet Vision, um sistema embarcado voltado ao registro fotográfico automatizado das visitas de felinos domésticos ao bebedouro (Felis catus). A solução visa auxiliar tutores e médicos-veterinários no acompanhamento preventivo e na detecção precoce de patologias como as Doenças do Trato Urinário Inferior Felino (DTUIF) e a Doença Renal Crônica (DRC). O projeto de hardware e arquitetura emprega como unidade central a placa ESP32-CAM (sensor OV2640 em resolução QVGA), integrada a um expansor de entradas e saídas PCF8574 via barramento I2C compartilhado com o relógio em tempo real RTC DS1307. A Interface Homem-Máquina (IHM) foi concebida para exibição em display TFT ILI9341 de 2,4 polegadas. O sensoriamento de presença opera por meio do sensor ultrassônico HC-SR04, acionado de forma não bloqueante por interrupção de hardware na GPIO 3 e validado por algoritmo de leituras consecutivas para eliminação de falsos positivos. Para a operação em ambientes de baixa luminosidade, especificou-se uma matriz de seis LEDs infravermelhos (850 nm) chaveada por transistor MOSFET e controlada automaticamente por um fotorresistor (LDR) com estabilização temporal. O firmware foi integralmente desenvolvido em linguagem C++ sob temporização assíncrona baseada em interrupções e contadores de tempo (millis()), favorecendo o processamento cooperativo dos periféricos, com apenas pausas bloqueantes pontuais e documentadas. O fluxo operacional prevê que, a cada visita confirmada, o sistema realize a captura fotográfica em formato JPEG (com limite de fotos por visita), armazene as imagens no cartão MicroSD e registre a estampa temporal de cada foto e o resumo de cada visita em arquivos locais CSV, salvando parâmetros operacionais em memória não volátil (NVS). Por meio de análise estática do código-fonte, rastreamento lógico de cenários operacionais, cálculos de engenharia elétrica e simulação conceitual, verificou-se, de forma analítica, a coerência técnica e a viabilidade elétrica e temporal da plataforma e estimou-se o custo de materiais. O protótipo físico não foi montado; portanto, taxa de detecção, qualidade das imagens e consumo reais não foram medidos e constam de um plano de ensaios. O trabalho entrega uma arquitetura modular, de custo estimado reduzido, preparada para implementações físicas futuras e expansões online no setor de tecnologia voltado ao bem-estar animal (pet tech).

Palavras-chave: Módulo de Câmera ESP32; Sistemas Embarcados; Internet das Coisas; Monitoramento Felino; Registro Fotográfico Noturno.

ABSTRACT

This work presents the design and analytical verification of version 2.0 of HydraPet Vision, an embedded system for the automated photographic logging of domestic cats' visits (Felis catus) to a water bowl, intended to support owners and veterinarians in the preventive follow-up of conditions such as feline lower urinary tract disease and chronic kidney disease. The design is based on an ESP32-CAM board (OV2640 sensor, QVGA), a PCF8574 I2C expander shared with a DS1307 real-time clock, a 2.4-inch ILI9341 TFT display, an HC-SR04 ultrasonic sensor read through a hardware interrupt, and an array of six 850 nm infrared LEDs switched by a MOSFET and enabled by an LDR module with time filtering. The firmware, written in C++ for the Arduino IDE, uses cooperative scheduling with a few documented blocking pauses. Each confirmed visit is logged with start time, duration and number of photos, and each JPEG photo is stored on a MicroSD card with its record in CSV files; settings persist in non-volatile storage (NVS). Verification relied on static code analysis, engineering calculations and conceptual scenario simulation. The physical prototype was not assembled, so detection rate, image quality and measured power consumption remain unknown; a bench-test plan with predefined acceptance criteria is provided. The work delivers a modular design with an estimated low material cost and a roadmap for physical implementation and future connected expansion.

Keywords: ESP32-CAM; Embedded Systems; Internet of Things; Feline Monitoring; Night-time Photographic Logging.

LISTA DE ILUSTRAÇÕES

Figura 1 – Quantidade de gatos que os tutores possuem. 29
Figura 2 – Dados quanto aos gatos viverem dentro ou fora de casa. 30
Figura 3 – Faixa etária dos gatos. 30
Figura 4 – Costume de levar os gatos ao veterinário para consultas preventivas. 31
Figura 5 – Informações se os tutores sabem o quanto os gatos bebem de água por dia. 31
Figura 6 – Costume de observar se o gato bebe água. 32
Figura 7 – Nível de preocupação dos tutores quanto ao gato estar bebendo pouca água. 32
Figura 8 – Nível de preocupação dos tutores quanto ao gato estar bebendo muita água 33
Figura 9 – Monitoramento dos tutores quanto ao gato estar bebendo água em suas ausências. 33
Figura 10 – Problemas apontados em relação à água disponível para os felinos. 34
Figura 11 – Maior preocupação dos tutores quanto à água quando estão fora de casa. 35
Figura 12 – Utilização de dispositivo inteligente para o gato. 35
Figura 13 – Interesse do público-alvo em receber informações sobre o consumo de água pelos gatos. 36
Figura 14 – Informações que os cuidadores consideram mais úteis quanto à hidratação dos gatos. 36
Figura 15 – Nível de interesse dos cuidadores em utilizar o sistema conforme proposto no projeto. 37
Figura 16 – Funcionalidade no sistema que os tutores consideram ser mais importante 38
Figura 17 – Utilidade de câmera de monitoramento integrada ao dispositivo segundo os tutores. 38
Figura 18 – Possibilidade de adquirir um dispositivo que possa ser acoplado ao bebedouro existente segundo os cuidadores. 39
Figura 19 – Valor considerado aceitável pelos entrevistados para pagar no dispositivo. 39
Figura 20 – Possibilidade de pagar mensalidade para o serviço de monitoramento segundo os tutores. 40
Figura 21 – Preferência dos tutores em relação às versões básica sem mensalidade ou premium com assinatura mensal para o serviço de monitoramento. 40
Figura 22 – Situações consideradas para avaliar a utilidade de um sistema de monitoramento de água segundo os tutores. 41

Figura 23 – Dados em relação ao possível alerta da aplicação quanto à mudança no padrão de consumo de água pelo gato e atitudes resultantes dos tutores. 41
Figura 24 – Arquitetura funcional do protótipo HydraPet Vision 2.0. 51
Figura 25 – Vistas frontal, superior e lateral do protótipo HydraPet Vision 2.0, com identificação dos principais elementos. 52
Figura 26 – Concepção ilustrativa do protótipo HydraPet Vision 2.0 em operação junto ao recipiente de água. 53
Figura 27 – ESP32-CAM 54
Figura 28 – Display TFT ILI9341 56
Figura 29 – PCF8574: Módulo expansor de portas I2C 58
Figura 30 – Sensor ultrassônico HC-SR04 59
Figura 31 – Módulo Sensor de luz LDR, com saídas analógica e digital 60
Figura 32 – LEDs infravermelhos de 850 nm 61
Figura 33 – Módulo RTC DS1307 62
Figura 34 – Cartão MicroSD 63
Figura 35 – Fluxograma do tratamento dos eventos dos botões pelo firmware do HydraPet Vision 2.0. 79
Figura 36 – Fluxograma de navegação entre as telas da interface homem-máquina do HydraPet Vision 2.0. 80
Figura 37 – Telas T01 a T10 da IHM: inicialização, tela inicial, menu e início do ajuste de data e hora. 86
Figura 38 – Telas T11 a T20 da IHM: conclusão do ajuste de data e hora e início da configuração dos parâmetros. 88
Figura 39 – Telas T21 a T30 da IHM: modos do infravermelho, demais parâmetros e aproximação do animal. 90
Figura 40 – Telas T31 a T40 da IHM: capturas com luz normal e infravermelha, contadores e lista de registros. 92
Figura 41 – Telas T41 a T45 da IHM: lista de registros, galeria de fotografias e informações do sistema. 94
Figura 42 – Fluxo lógico de funcionamento do firmware HydraPet Vision v2.0. 97
Figura 43 – Montagem experimental do HydraPet Vision 2.0 sobre protoboard, com identificação dos módulos. 102
Figura 44 – Detalhamento do protótipo em padrão de desenho técnico industrial: vistas frontal, lateral direita e isométrica 103

Figura 45 – Diagrama elétrico completo do protótipo HydraPet Vision v2.0 105
Figura 46 – Composição estimada da latência de captura e registro 122
Figura 47 – Composição estimada da corrente por estado de operação 123
Figura 48 – Exemplo ilustrativo do conteúdo de registros.csv (dados hipotéticos) 124

LISTA DE TABELAS

Tabela 1 – Comparativo entre o HydraPet Vision 2.0 e projetos correlatos desenvolvidos com arquiteturas similares. 44
Tabela 2 – Mapeamento principal de pinos da ESP32-CAM no firmware 2.0 54
Tabela 3 – Pinagem complementar do sensor de imagem OV2640 na ESP32-CAM 55
Tabela 4 – Distribuição das portas do PCF8574 57
Tabela 5 – Evolução funcional entre a versão inicial e a versão 2.0. 106
Tabela 6 – Níveis de verificação adotados e alcance de cada um 111
Tabela 7 – Rastreabilidade entre requisitos, firmware e verificação 114
Tabela 8 – Largura de pulso do sinal ECHO esperada para distâncias representativas 117
Tabela 9 – Tempo mínimo de validação em função do número de leituras de confirmação 118
Tabela 10 – Modos de iluminação infravermelha e comportamento esperado 120
Tabela 11 – Estimativa de latência do ciclo de captura e registro 122
Tabela 12 – Orçamento de consumo estimado do protótipo (alimentação de 5 V) 123
Tabela 13 – Capacidade estimada de um cartão MicroSD de 16 GB 125
Tabela 14 – Cenários simulados e comportamento esperado do firmware 126
Tabela 15 – Riscos técnicos identificados e atenuação 128
Tabela 16 – Custo estimado de materiais do protótipo HydraPet Vision v2.0 129
Tabela 17 – Roteiro de ensaios para validação física do protótipo 131
Tabela 18 – Roteiro de evolução: versão 2.0 (offline) e versão 3.0 (conectada) 133

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AEC	Auto Exposure Control (Controle Automático de Exposição)
AI	Artificial Intelligence (Inteligência Artificial)
AM	Ante Meridiem
C++	Linguagem de Programação C++
CI	Circuito Integrado
CS	Chip Select
CSV	Comma-Separated Values (Valores Separados por Vírgula)
DC	Data/Command (Dados/Comando)
DMA	Direct Memory Access (Acesso Direto à Memória)
DRC	Doença Renal Crônica
DTUIF	Doença do Trato Urinário Inferior Felino
DVP	Digital Video Port
E/S	Entrada e Saída
ECHO	Echo (Sinal de Retorno Ultrassônico)
ESP-IDF	Espressif IoT Development Framework
FDM	Fused Deposition Modeling (Modelagem por Deposição Fundida)
FS	File System (Sistema de Arquivos)
FW	Firmware
GPIO	General Purpose Input/Output (Entrada/Saída de Propósito Geral)
GND	Ground (Terra / Referência Zero Volt)
HREF	Horizontal Reference (Referência Horizontal de Vídeo)
HSPI	Hardware Serial Peripheral Interface
I2C	Inter-Integrated Circuit

IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
IHM	Interface Homem-Máquina
IoT	Internet of Things (Internet das Coisas)
IR	Infrared (Infravermelho)
IRAM	Internal Static Random-Access Memory (Memória RAM Interna em Código Executável)
ISR	Interrupt Service Routine (Rotina de Serviço de Interrupção)
JPEG	Joint Photographic Experts Group
LDR	Light Dependent Resistor (Resistor Dependente de Luz / Fotorresistor)
LED	Light Emitting Diode (Diodo Emissor de Luz)
MCU	Minimum Coded Unit (Unidade Mínima Codificada)
MicroSD	Micro Secure Digital
MISO	Master In Slave Out
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MOSI	Master Out Slave In
MQTT	Message Queuing Telemetry Transport
NVS	Non-Volatile Storage (Armazenamento Não Volátil)
OV	OmniVision (Fabricante do Sensor de Imagem OV2640)
PCF	Philips CMOS Family (Família de CIs Expansores I2C)
PM	Post Meridiem
PSRAM	Pseudo Static Random-Access Memory (Memória RAM Pseudo-Estática External)
QVGA	Quarter Video Graphics Array (320x240 Pixels)
RAM	Random Access Memory (Memória de Acesso Aleatório)
RGB	Red, Green, Blue (Vermelho, Verde e Azul)
RST	Reset

RTC	Real-Time Clock (Relógio em Tempo Real)
SCCB	Serial Camera Control Bus
SCK	Serial Clock
SD	Secure Digital
SDI	Serial Data In
SDMMC	Secure Digital Multimedia Card Controller
SoC	System on Chip (Sistema em um Chip)
SPI	Serial Peripheral Interface
TFT	Thin-Film Transistor
TRIG	Trigger (Gatilho de Disparo Ultrassônico)
UART	Universal Asynchronous Receiver-Transmitter
VSYNC	Vertical Synchronization (Sincronismo Vertical de Vídeo)
Wi-Fi	Wireless Fidelity
XCLK	External Clock (Clock Externo da Câmera)

Sumário

1. INTRODUÇÃO 17

1.1 Contextualização e Problema 18

1.2 Objetivos 20

1.2.1 Objetivo Geral 20

1.2.2 Objetivos Específicos 20

1.3 Nomenclatura e Propósito do Projeto – HydraPet Vision 21

1.4 Estrutura do Trabalho e Resumo Geral dos Capítulos 22

2 FUNDAMENTAÇÃO TEÓRICA E METODOLOGIA 25

2.1 A importância da hidratação para os gatos domésticos 25

2.2 A Engenharia de Sistemas Embarcados e IoT Aplicada à Saúde Preventiva 26

2.3 Metodologia 26

2.3.1 Entrevistas e levantamento de requisitos com tutores de gatos 26

2.3.2 Arquitetura geral do protótipo 27

2.4 Resultados Preliminares: Solução Inicial 28

2.4.1 Resultados do levantamento com o público-alvo 28

2.5 Resultados Preliminares da Solução Inicial 42

2.6 Trabalhos Correlatos e Comparativo Tecnológico 43

2.7 A Natureza IoT do Sistema Frente à Operação Autônoma e Offline 46

3 DESENVOLVIMENTO DO PROTÓTIPO 48

3.1 Arquitetura Geral do Protótipo 49

3.2 Detalhamento do Hardware e Conexões 53

3.2.1 Mapeamento de pinos da ESP32-CAM 54

3.2.2 Display TFT ILI9341 55

3.2.3 Barramento I2C compartilhado 57

3.2.4 Expansor I/O PCF8574 57

3.2.5 Sensor ultrassônico HC-SR04 58

3.2.6 Sensor LDR e detecção de luminosidade 59

3.2.7 Iluminação infravermelha de 850 nm 60
3.2.8 Módulo RTC DS1307 61
3.2.9 Cartão MicroSD e armazenamento local 62
3.3 Desenvolvimento do Firmware 63
3.3.1 Biblioteca Arduino.h 63
3.3.2 Biblioteca Wire.h 65
3.3.3 Biblioteca SPI.h 65
3.3.4 Biblioteca Adafruit_GFX.h 66
3.3.5 Biblioteca Adafruit_ILI9341.h 66
3.3.6 Biblioteca RTClib.h 66
3.3.7 Biblioteca TJpg_Decoder.h 67
3.3.8 Biblioteca Preferences.h 67
3.3.9 Biblioteca driver/gpio.h 68
3.3.10 Biblioteca esp_camera.h 68
3.3.11 Bibliotecas FS.h e SD_MMC.h 69
3.3.12 Inicialização do sistema 69
3.3.13 Configuração persistente por NVS 70
3.4 Sensoriamento de Presença e Confirmação do Evento 71
3.5 Controle Automático da Iluminação Infravermelha 72
3.6 Processo de Captura Fotográfica 74
3.7 Banco de Registros e Organização dos Dados 75
3.8 Interface Homem-Máquina 77
3.8.1 Tela inicial 81
3.8.2 Menu principal 82
3.8.3 Registros de fotos 82
3.8.4 Galeria de fotografias 83
3.8.5 Ajuste de data e hora 83

3.8.6 Configurações do sistema 84
3.8.7 Tela de informações do sistema 85
3.8.8 Sequência completa das telas da interface 85
3.9 Bibliotecas e Funções Principais do Firmware 95
3.10 Máquina de Estados e Organização do Fluxo Operacional 96
3.11 Tratamento de Falhas e Recuperação do Sistema 97
3.12 Estratégias de Temporização e Redução de Bloqueios 98
3.13 Fluxo Completo de Funcionamento do HydraPet Vision 2.0 99
3.14 Integração Entre Hardware e Software 100
3.15 Considerações Sobre a Montagem e Programação do Protótipo 100
3.15.1 Montagem experimental em protoboard 101
3.15.2 Invólucro e vistas de detalhamento 102
3.16 Diagrama Elétrico e Correspondência com o Firmware 104
3.17 Síntese da Evolução Entre as Versões 105
3.18 Síntese do Funcionamento Técnico 106
4 RESULTADOS E DISCUSSÕES 110
4.1 Estratégia de Verificação 110
4.1.1 Natureza da avaliação e delimitação do alcance 111
4.1.2 Critérios de avaliação 111
4.2 Verificação da Arquitetura de Hardware 112
4.2.1 Rastreabilidade dos requisitos 112
4.2.2 Alocação de pinos e barramento I2C compartilhado 115
4.2.3 Compatibilidade de níveis lógicos do sinal ECHO 115
4.2.4 Dimensionamento do circuito de iluminação infravermelha 116
4.3 Sensoriamento de Proximidade e Critério de Confirmação 116
4.3.1 Relação entre distância e largura do pulso 117
4.3.2 Influência da temperatura 117

4.3.3 Tempo de validação e escolha do número de leituras 118
4.4 Iluminação Infravermelha e Qualidade de Imagem Esperada 119
4.4.1 Estratégia de acionamento sob demanda 119
4.4.2 Condicionantes ópticos 120
4.5 Desempenho Temporal e Consumo de Energia 120
4.5.1 Ciclo Cooperativo e Responsividade 121
4.5.2 Latência do ciclo de detecção ao registro 121
4.5.3 Orçamento de consumo e alimentação 122
4.6 Registro, Armazenamento e Banco de Dados Local 124
4.6.1 Estrutura dos dados persistidos 124
4.6.2 Capacidade de armazenamento 125
4.6.3 Integridade dos registros 125
4.7 Simulação Conceitual de Cenários de Uso 126
4.8 Riscos Técnicos e Medidas de Atenuação 128
4.9 Levantamento de Custos de Materiais 129
4.10 Plano de Validação Experimental 131
4.11 Discussão Geral e Perspectivas de Expansão 132
4.12 Síntese do Capítulo 133
5. CONCLUSÃO 136
5.1 Síntese das Conquistas e Resultados da Versão 2.0 137
5.2 Trabalhos e Evoluções Futuras 138
5.3 Considerações finais 139
REFERÊNCIAS BIBLIOGRÁFICAS 140
APÊNDICE A – CÓDIGO-FONTE DO FIRMWARE E EVIDÊNCIA DE COMPILAÇÃO 144
A.1 Disponibilização do código-fonte 144
A.2 Ambiente de compilação 144

A.3 Evidência de compilação 145

1. INTRODUÇÃO

A convivência e o papel dos animais de estimação na estrutura familiar contemporânea passaram por transformações profundas nas últimas décadas. Em especial, a população de felinos domésticos (*Felis catus*) apresentou um crescimento expressivo em ambientes urbanos, motivado por características como o comportamento independente e a facilidade de adaptação a espaços reduzidos. Contudo, essa mudança de ecossistema e estilo de vida trouxe desafios significativos à saúde e à longevidade desses animais. O confinamento domiciliar, aliado ao consumo predominante de dietas secas extrusadas e ao sedentarismo, diverge drasticamente dos hábitos alimentares de seus ancestrais, o que pode resultar em ingestão hídrica insuficiente (ZANGHI; GERHEART; GARDNER, 2018; ZORAN; BUFFINGTON, 2011).

A desidratação crônica de baixo grau é uma condição silenciosa e recorrente em felinos, podendo contribuir para o surgimento de patologias renais e urinárias graves, como a Doença Renal Crônica (DRC) e a Doença do Trato Urinário Inferior Felino (DTUIF) (POLZIN, 2011; SPARKES et al., 2016). No contexto da medicina veterinária preventiva, o acompanhamento rigoroso do padrão e da frequência do consumo de água configura-se como um dos indicadores clínicos mais relevantes para o diagnóstico precoce dessas enfermidades (SPARKES et al., 2016; LITTLE, 2011). Todavia, a mensuração precisa desse parâmetro no ambiente residencial esbarra em limitações práticas, visto que a rotina dos tutores impede a observação contínua e a mensuração manual de volume não isola fatores ambientais, como a evaporação natural do recipiente de água ou o acesso de múltiplos animais coabitantes.

Diante dessa problemática, a aplicação de recursos tecnológicos da Engenharia da Computação e dos Sistemas Embarcados surge como uma alternativa viável para automatizar a coleta de dados e auditar o comportamento animal de forma não invasiva (BAHGA; MADISETTI, 2014; LEE; SESHIA, 2017). O projeto *HydraPet Vision* foi concebido precisamente com essa finalidade, propondo o desenvolvimento, a especificação de *firmware* e a validação conceitual de uma plataforma embarcada de baixo custo capaz de monitorar, registrar e auditar os eventos de hidratação felina. Desenvolvido no âmbito acadêmico como um projeto estruturado em nível de especificação de hardware, codificação em linguagem C++ concebida e compilada no ambiente integrado Arduino IDE (utilizando o núcleo ESP32 fornecido pela *Espressif Systems*) e simulação lógica de cenários operacionais, sem a montagem física do protótipo, o sistema prioriza a alta confiabilidade no armazenamento local de dados e a independência de infraestrutura de rede, enquanto mantém uma arquitetura modular totalmente preparada para futuras implementações físicas e expansões *online*.

Ademais, o desenvolvimento deste trabalho cumpre um papel fundamental no âmbito acadêmico e profissional, atuando como um instrumento de consolidação dos conhecimentos multidisciplinares adquiridos ao longo do curso de Engenharia de Computação, abrangendo áreas como eletrônica aplicada, desenvolvimento de *firmware*, arquitetura de software para microcontroladores, projetos de interface homem-máquina e engenharia de software (LEE; SESHIA, 2017; MONK, 2016). Por fim, o projeto é estruturado com o propósito de demonstrar a viabilidade técnica e mercadológica de um dispositivo que, mediante posterior montagem e aprimoramentos físicos, apresenta potencial concreto para ser produzido e comercializado no setor de tecnologia voltado ao bem-estar animal (*pet tech*).

1.1 Contextualização e Problema

Os felinos domésticos (*Felis catus*), originados do gato selvagem do Oriente Próximo (DRISCOLL et al., 2007), costumam apresentar ingestão espontânea de água relativamente baixa e urina concentrada, embora a ideia de um "ancestral do deserto" seja questionada (ZORAN; BUFFINGTON, 2011). Embora essa característica tenha se configurado como uma adaptação evolutiva eficiente para a sobrevivência em ambientes de escassez hídrica, ela representa um expressivo fator de risco na domesticidade moderna. A transição do regime alimentar ancestral, baseado em presas com alto teor de água, para dietas secas extrusadas reduz o teor de água da dieta e exige maior ingestão voluntária de água (ZANGHI; GERHEART; GARDNER, 2018). A menor diluição da urina favorece a saturação de solutos (BUCKLEY et al., 2011), o que se associa a maior risco de Doenças do Trato Urinário Inferior Felino (DTUIF) ; a Doença Renal Crônica (DRC), por sua vez, é comum em gatos idosos (MARINO et al., 2014; SPARKES et al., 2016).

No âmbito clínico veterinário, o monitoramento contínuo da frequência e do padrão comportamental de ingestão de água constitui um biomarcador não invasivo de extrema relevância para a detecção precoce de anomalias que antecedem quadros clínicos severos (LITTLE, 2011; DIBARTOLA, 2012). Contudo, a medição precisa do consumo hídrico individual apresenta severas limitações práticas no ambiente domiciliar. A medição indireta por variação de volume do recipiente falha em isolar fatores de evaporação natural ou a interferência de múltiplos animais coabitantes. Por sua vez, a observação direta por parte dos tutores revela-se inviável devido a rotinas de trabalho e ao comportamento essencialmente noturno e reservado da espécie.

Embora existam soluções comerciais voltadas ao bem-estar pet, parte delas depende de aplicativo e de conexão de rede e, em alguns casos, de assinatura, com custo mais elevado (PETLIBRO, [s.d.]; PETKIT, [s.d.]; SURE PETCARE, [s.d.]). Ademais, sob a perspectiva da auditabilidade de dados, nos produtos e projetos consultados (Seção 2.6) nem sempre há registro fotográfico com operação noturna documentada, comprometendo o registro fidedigno do evento de consumo hídrico.

Nesse cenário, o presente projeto surge como uma solução tecnológica acessível e eficiente concebida em nível de *firmware* e validação lógica. Por se tratar do desenvolvimento de uma especificação avançada e prova de conceito computacional, estruturada por meio de programação em C++ no ambiente Arduino IDE e análise conceitual de desempenho, visto que o protótipo físico não foi montado, o dispositivo foi arquitetado para operar de maneira prioritariamente *offline*, com armazenamento local dos dados e independência de redes sem fio, o que elimina a dependência de conexão (a confiabilidade real depende dos ensaios).

Para a concretização das funcionalidades lógicas do sistema, o desenvolvimento do código-fonte fundamenta-se na utilização de bibliotecas nativas e de terceiros altamente otimizadas para microcontroladores (MONK, 2016). O ecossistema de software integra as bibliotecas padrão do núcleo ESP32 e componentes avançados disponibilizados pela Espressif Systems ([s.d.a]) (fabricante do SoC ESP32-CAM), tais como driver/gpio.h (para o gerenciamento de interrupções de hardware de baixo nível e controle preciso de pinos) e esp_camera.h (para a interface direta com o sensor de imagem OV2640, controle de DMA e gestão de *framebuffers* em PSRAM), além das bibliotecas de sistema de arquivos FS.h e SD_MMC.h (responsáveis pela comunicação em barramento SDMMC de 1 bit com o cartão MicroSD) e Preferences.h (para a gestão da memória não volátil NVS).

Ademais, o *firmware* emprega bibliotecas consolidadas da comunidade e de fabricantes para o suporte aos periféricos externos, destacando-se a Adafruit_GFX (ADAFRUIT INDUSTRIES, [s.d.]) e a Adafruit_ILI9341 (ADAFRUIT INDUSTRIES, [s.d.]) (para a renderização de primitivas gráficas e controle do display TFT via interface SPI de alta velocidade), a RTClib (desenvolvida pela *Adafruit* para a interface com o relógio em tempo real DS1307 via barramento I2C) (ANALOG DEVICES, 2015), a TJpg_Decoder (desenvolvida por *Bodmer* para a descompressão e exibição otimizada de imagens JPEG na galeria local) e as bibliotecas nativas Wire.h e SPI.h para a gerência dos barramentos de comunicação síncrona.

Não obstante, a arquitetura de hardware e software foi concebida de forma modular, prevendo a facilidade de integração em atualizações futuras para conectividade *online* (nuvem, comunicação MQTT/HTTP e monitoramento via aplicativo *mobile*) (BAHGA; MADISETTI,

2014). Além da relevância clínica e prática no manejo felino, a concepção deste projeto tem como objetivo fundamental integrar e demonstrar na prática os conhecimentos multidisciplinares adquiridos ao longo da graduação, contemplando sistemas embarcados, processamento de imagens (GONZALEZ; WOODS, 2018), eletrônica aplicada, engenharia de software e arquitetura de código. A abordagem adotada busca validar a viabilidade técnica e financeira de um produto inovador, estruturado desde o seu design inicial para ser comercializado no futuro a um custo competitivo para o mercado pet.

1.2 Objetivos

A presente seção estabelece a meta central deste projeto integrador e desdobra o seu escopo em etapas técnicas essenciais. Apresentam-se a seguir o objetivo geral, que delineia a entrega principal do projeto, e os objetivos específicos, que organizam metodologicamente os passos necessários para a concepção, o desenvolvimento do *firmware* no ambiente Arduino IDE e a validação conceitual do sistema embarcado.

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é projetar, desenvolver o código-fonte (*firmware*) na plataforma Arduino IDE e documentar a versão 2.0 de um sistema embarcado de baixo custo e operação *offline* (com arquitetura preparada para futuras expansões *online* e validação conduzida por meio de simulação conceitual e análise lógica, sem a montagem de protótipo físico), baseado na placa ESP32-CAM, capaz de monitorar de forma não invasiva a hidratação felina. O sistema foi programado para detectar a aproximação do animal por sensoriamento de proximidade, realizar a validação lógica e o registro fotográfico automático da visita (inclusive em ambientes noturnos via iluminação infravermelha), indexar os registros com registros precisos de data e hora (*timestamping*) em um banco de dados estruturado e disponibilizar os dados e imagens através de uma Interface Homem-Máquina (IHM) interativa em display TFT, demonstrando a viabilidade de um dispositivo apto para futura montagem e comercialização.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral proposto, foram definidos objetivos específicos que orientam o desenvolvimento lógico do protótipo *HydraPet Vision 2.0* e abrangem os principais aspectos de hardware, *firmware*, interface, armazenamento de dados e estrutura lógica do sistema. Inicialmente, busca-se projetar a integração do sensor ultrassônico HC-SR04 de forma

eficiente e não bloqueante, utilizando a leitura do sinal de eco orientada por interrupções de hardware por meio de *attachInterrupt* com o auxílio das rotinas de baixo nível da *Espressif* em driver/gpio.h, associada a um mecanismo programável de confirmação de presença por leituras consecutivas para eliminação de falsos positivos (TOCCI; WIDMER; MOSS, 2007). Em complemento, o projeto especifica a automação da iluminação noturna mediante a utilização de um sistema de LEDs infravermelhos de 850 nm acionados por transistor MOSFET e controlados automaticamente por um fotorresistor LDR com estabilização temporal, com o objetivo de obter capturas nítidas em ambientes de baixa luminosidade (a confirmar em ensaio).

Adicionalmente, visa-se otimizar a utilização das portas de entrada e saída da ESP32-CAM por meio da adoção do expansor I2C PCF8574, concentrando o acionamento do ultrassom, a leitura do LDR, o controle do infravermelho e a varredura dos botões de navegação, o que preserva as linhas dedicadas à câmera OV2640 e ao cartão MicroSD. No que tange à interação com o usuário, o projeto contempla o desenvolvimento de uma Interface Homem-Máquina (IHM) interativa em display TFT ILI9341 de 2,4 polegadas, estruturada com o suporte das bibliotecas gráficas da *Adafruit* e do decodificador JPEG de *Bodmer*, viabilizando a navegação por menus, a visualização da galeria de fotografias, o ajuste de data e hora no relógio de tempo real (RTC) e a verificação de diagnósticos do sistema.

A persistência de dados e metadados foi programada para ocorrer de forma estruturada, implementando a gravação de registros tabulares em formato CSV no cartão MicroSD com o auxílio das bibliotecas FS.h e SD_MMC.h, juntamente com a preservação de parâmetros operacionais em memória não volátil NVS por meio da biblioteca Preferences.h. Por fim, o projeto compreende a concepção mecânica modular do dispositivo, consistindo na especificação de um invólucro compacto estruturado por manufatura aditiva através do processo de impressão 3D FDM, de modo a garantir o alinhamento óptico adequado e a proteção física dos componentes para a operação no ambiente doméstico.

1.3 Nomenclatura e Propósito do Projeto – HydraPet Vision

O nome *HydraPet Vision* resulta da síntese de conceitos fundamentais que expressam a proposta biotecnológica e funcional do projeto. O termo *Hydra* remete ao vocábulo grego *Hydor* (água), fazendo referência direta ao objetivo central da pesquisa: incentivar, monitorar e assegurar a adequada hidratação felina. Essa atenção ao consumo hídrico revela-se indispensável para a prevenção de enfermidades graves, tais como a Doença Renal Crônica (DRC) e as Doenças do Trato Urinário Inferior Felino (DTUIF) (SPARKES et al., 2016).

O vocábulo *Pet* delimita o público-alvo da aplicação, compreendendo os animais de companhia. Por sua vez, a palavra *Vision* destaca a captura fotográfica automatizada e o sensoriamento noturno por radiação infravermelha, que permitem a conferência visual dos eventos de aproximação ao bebedouro (a análise automática das imagens, isto é, a visão computacional propriamente dita, é uma evolução futura).

Complementando a identidade do projeto, o subtítulo *"Tecnologia a serviço do bem-estar animal"* reforça a premissa ética da engenharia aplicada. A proposta balizadora fundamenta-se em empregar os avanços da Internet das Coisas (IoT) e dos sistemas embarcados em favor da saúde preventiva, promovendo o incremento da qualidade de vida dos felinos domésticos.

1.4 Estrutura do Trabalho e Resumo Geral dos Capítulos

Esta seção apresenta uma visão geral da organização do trabalho e resume o conteúdo de cada capítulo, com o objetivo de orientar a leitura e de relacionar o problema, o desenvolvimento e a verificação do HydraPet Vision 2.0.

O Capítulo 1 contextualiza o problema: a baixa ingestão espontânea de água pelos gatos domésticos, a dificuldade de tutores e médicos-veterinários acompanharem esse hábito e a relação entre hidratação e doenças como a Doença Renal Crônica (DRC) e as Doenças do Trato Urinário Inferior Felino (DTUIF). A partir disso, a Seção 1.2 define como objetivo geral projetar, programar e documentar um sistema embarcado de baixo custo e operação offline, baseado na ESP32-CAM, que detecta a aproximação do animal ao bebedouro e registra a visita por fotografia, inclusive à noite, e o desdobra em objetivos específicos de sensoriamento, iluminação infravermelha, interface local e armazenamento. A Seção 1.3 explica a origem do nome do projeto, e esta Seção 1.4 descreve a estrutura do documento.

O Capítulo 2 reúne a fundamentação teórica e a metodologia. Trata da fisiologia hídrica do gato e da importância de monitorar a hidratação (Seção 2.1), do papel dos sistemas embarcados e da Internet das Coisas na saúde preventiva (Seção 2.2) e da metodologia adotada (Seção 2.3), que inclui o levantamento de requisitos com tutores e a arquitetura geral do protótipo. A pesquisa online respondida por 98 tutores (Seção 2.4) indicou interesse expressivo no dispositivo, boa receptividade a um módulo acoplável ao bebedouro já existente e, como faixa de preço mais citada, de R$ 100,00 a R$ 200,00, ressalvando-se que se trata de amostra de conveniência. O capítulo também posiciona o projeto frente a trabalhos acadêmicos e

produtos comerciais (Seção 2.6) e justifica seu enquadramento como nó de borda de IoT, de operação autônoma e offline na versão atual (Seção 2.7).

O Capítulo 3 descreve o desenvolvimento do protótipo. Apresenta a arquitetura modular em torno da ESP32-CAM (Seção 3.1) e o detalhamento do hardware (Seção 3.2): mapeamento de pinos, display TFT ILI9341, barramento I2C compartilhado, expansor PCF8574, sensor ultrassônico HC-SR04, sensor de luminosidade LDR, matriz de seis LEDs infravermelhos de 850 nm, relógio DS1307 e cartão MicroSD. Em seguida, detalha o firmware em C++ para o Arduino IDE (Seção 3.3), com as bibliotecas, a inicialização e a configuração persistente em NVS, e as rotinas de sensoriamento de presença por interrupção com confirmação por leituras consecutivas (Seção 3.4), de controle automático do infravermelho (Seção 3.5), de captura fotográfica (Seção 3.6) e de registro em arquivos CSV (Seção 3.7). Descreve ainda a interface homem-máquina (Seção 3.8), a máquina de estados, o tratamento de falhas e as estratégias de temporização (Seções 3.10 a 3.12), o fluxo completo de funcionamento e a integração entre hardware e software (Seções 3.13 e 3.14), as considerações de montagem e o diagrama elétrico (Seções 3.15 e 3.16), a evolução em relação à versão inicial (Seção 3.17) e a síntese do funcionamento (Seção 3.18). A Seção 3.19 registra a revisão 2.1 do firmware, que adota a visita como unidade de registro, organiza as fotografias por data e diagnostica falhas do sensor ultrassônico, e que foi compilada sem erros, ainda sem execução em hardware.

O Capítulo 4 apresenta os resultados e a discussão. Como o protótipo físico não foi montado, a verificação é analítica e conceitual (Seção 4.1), baseada em análise estática do código, cálculos de engenharia e rastreamento lógico de cenários. Os resultados indicam que os treze requisitos definidos têm mecanismo correspondente no firmware; que o divisor resistivo do sinal ECHO fornece cerca de 3,33 V, exigindo resistores de 1% no pior caso, e que a matriz infravermelha drena cerca de 66 mA (Seção 4.2); que a confirmação por quatro leituras consecutivas consome 180 ms e que a latência estimada entre a detecção e o registro concluído é inferior a um segundo (Seções 4.3 e 4.5); que o consumo estimado varia de 255 mA a 481 mA a 5 V, com margem de cerca de quatro vezes na fonte de 2 A (Seção 4.5); e que a capacidade do cartão MicroSD de 16 GB não é limitante (Seção 4.6). O capítulo traz ainda a simulação de nove cenários de uso (Seção 4.7), os riscos técnicos e suas medidas de atenuação, como o uso de MOSFET de nível lógico e de câmera sem filtro de infravermelho (Seção 4.8), o custo de materiais de R$ 285,40 (Seção 4.9), o plano de validação experimental, com critérios de aceitação definidos previamente (Seção 4.10), e a discussão geral com as perspectivas de expansão (Seções 4.11 e 4.12). Delimita-se que o dispositivo registra eventos de aproximação e sua evidência fotográfica, e não o volume de água ingerido.

O Capítulo 5 sintetiza as conquistas da versão 2.0 (Seção 5.1): o uso do expansor PCF8574 para contornar a escassez de GPIOs, a leitura não bloqueante do HC-SR04 por interrupção, a iluminação infravermelha integrada ao fluxo de captura, a interface autônoma em display TFT e o armazenamento local com persistência em CSV e NVS. Aponta como trabalhos futuros (Seção 5.2) a montagem do protótipo e a execução do plano de ensaios, a placa de circuito impresso, a identificação individual dos animais por visão computacional embarcada, a medição do volume ingerido com célula de carga, a alimentação de emergência por baterias e a conectividade sem fio. As considerações finais (Seção 5.3) destacam como principal contribuição a especificação e a codificação completas de um sistema integrado, validado por análise e simulação, que deixa uma base de código pronta para a montagem física.

Por fim, as Referências Bibliográficas listam as fontes citadas ao longo do texto. Em todo o documento, os valores de desempenho, consumo e custo são estimativas de projeto, e não medições, uma vez que a validação em hardware consta do plano de ensaios da Seção 4.10.

2 FUNDAMENTAÇÃO TEÓRICA E METODOLOGIA

A temática deste projeto reveste-se de grande relevância, levando-se em conta a vulnerabilidade dos gatos domésticos ao baixo consumo de água, e o consequente risco de desenvolvimento e progressão de doenças renais relacionadas. O desenvolvimento de um monitor inteligente de hidratação felina exige a convergência de diversas áreas do conhecimento, incluindo a aplicação da eletrônica embarcada e da visão computacional como ferramentas não invasivas para a aquisição de dados. A fundamentação teórica consolida as justificativas técnica, biológica e comercial do projeto.

2.1 A importância da hidratação para os gatos domésticos

A evolução biológica do gato doméstico (*Felis catus*) resultou em um animal com um perfil fisiológico altamente especializado e, ao mesmo tempo, vulnerável no que tange ao metabolismo hídrico. Descendentes do gato selvagem do Oriente Próximo (DRISCOLL et al., 2007), os gatos domésticos costumam apresentar ingestão espontânea de água relativamente baixa, ainda que a ideia de um "ancestral do deserto" seja discutida (ZORAN; BUFFINGTON, 2011). Adicionalmente, a ingestão de água por gatos domésticos esbarra em vários fatores etológicos. O ato de beber água envolve preferências sensoriais e percepções ambientais de segurança. A água estagnada, por exemplo, é instintivamente associada pelos felinos a fontes que não são frescas. Na natureza, fluxos de água corrente sinalizam frescor e pureza, motivo pelo qual muitos tutores relatam preferência dos gatos por torneiras abertas ou fontes dinâmicas, embora um estudo com gatos de laboratório não tenha encontrado efeito geral do tipo de bebedouro sobre a ingestão de água (ROBBINS et al., 2019). Somado a isso, os fatores estressores, tais como os ruídos altos ou desconhecidos, os movimentos bruscos, os objetos e locais novos, a aproximação de estranhos (humanos, gatos ou outros animais) em seu território e doenças, principalmente às relacionadas ao trato urinário, podem interferir no hábito de beber água (STELLA; CRONEY; BUFFINGTON, 2013). Outro ponto a se considerar em relação aos gatos domésticos é a prevalência de dietas secas na alimentação, que têm baixo teor de umidade, em geral em torno de 10% (valor típico informado em rótulos de rações comerciais).

É conhecida na literatura a elevada incidência da Doença Renal Crônica (DRC) e Doenças do Trato Urinário Inferior Felino (DTUIF), um termo guarda-chuva que abrange afecções como a cistite idiopática felina, a urolitíase (formação de cristais minerais na urina) e as obstruções uretrais em gatos (POLZIN, 2011; POLZIN, 2013; MARINO et al., 2014). No caso da urolitíase, a supersaturação de minerais na urina favorece a formação e o crescimento

dos cristais (BUCKLEY et al., 2011). A desidratação recorrente pode contribuir para a perda progressiva da funcionalidade dos néfrons e, consequentemente, para a progressão dessas doenças, resultando em alta morbidade e mortalidade, principalmente em gatos idosos (POLZIN, 2011; POLZIN, 2013; SPARKES et al., 2016).

Neste panorama, um dos desafios enfrentados por médicos veterinários e tutores é a dificuldade de monitorar com precisão o volume hídrico consumido pelos gatos. Dentre as dificuldades, destaca-se a taxa de evaporação natural da água pela exposição prolongada ao ambiente. Outro ponto a ser observado é o caso de residências com múltiplos animais, nas quais a medição torna-se ainda mais complexa, uma vez que a simples aferição da variação de nível não permite identificar qual animal ingeriu o líquido, a frequência ou a duração dos episódios de hidratação. Neste sentido, alterações no padrão de consumo de água, principalmente se forem sutis, podem passar despercebidas, dificultando o manejo da hidratação dos animais, a identificação de fatores estressores e preferências que estão afetando o comportamento de beber água e a ocorrência de possíveis patologias relacionadas ao sistema urinário.

2.2 A Engenharia de Sistemas Embarcados e IoT Aplicada à Saúde Preventiva

A interseção entre a engenharia de computação, a Internet das Coisas (IoT) e a medicina veterinária oferece caminhos inovadores para superar os limites da observação humana na rotina de cuidados com os animais. Sistemas embarcados modernos, equipados com microcontroladores de baixo custo e alta capacidade de processamento, viabilizam a automação de leituras e a aquisição de dados em tempo real no próprio ambiente doméstico. No contexto da hidratação felina, a captura de imagens no momento exato em que o animal se aproxima do recipiente de água permite correlacionar o evento físico à presença efetiva do felino.

2.3 Metodologia

Nesta seção, foi descrita a metodologia que envolve o levantamento de requisitos junto ao público-alvo e o desenvolvimento técnico e a arquitetura geral do protótipo HydraPet Vision.

2.3.1 Entrevistas e levantamento de requisitos com tutores de gatos

A definição do local para a realização deste Projeto Integrador foi guiada pela necessidade de entender o mercado pet em relação aos gatos domésticos, e a partir das informações levantadas, avaliar a viabilidade comercial do dispositivo em desenvolvimento. Inicialmente, o grupo mapeou duas frentes distintas de atuação: clínicas veterinárias e tutores

de gatos. Nas reuniões do grupo, ponderou-se a viabilidade técnica e o potencial de impacto social de cada opção. Após o primeiro contato com uma clínica veterinária, o grupo concluiu que as informações necessárias para o levantamento de requisitos e viabilidade comercial seriam melhor obtidas através dos tutores, que seriam os consumidores finais da solução. Foi realizada uma coleta de dados para um estudo exploratório de caráter dedutivo, que buscou entender, por meio da aplicação de questionário quali-quantitativo com perguntas fechadas, sobre a visão e as necessidades dos tutores quanto ao monitoramento da hidratação dos gatos. O levantamento de dados foi realizado por meio da ferramenta Google Formulário apenas com tutores de gatos.

2.3.2 Arquitetura geral do protótipo

A arquitetura do HydraPet Vision foi concebida de forma modular, tendo a ESP32-CAM AI-Thinker como unidade central de processamento. A placa concentra o processamento do firmware, o controle da câmera OV2640, o acesso ao cartão MicroSD, a comunicação com o display TFT e a interação com os periféricos externos. O firmware do protótipo HydraPet Vision foi construído na linguagem C++, utilizando o ecossistema e a estrutura de compilação do ambiente Arduino IDE voltado à arquitetura de microcontroladores ESP32.

A arquitetura pode ser dividida em cinco subsistemas principais: processamento e captura de imagem, sensoriamento de presença, controle de iluminação, interface homem-máquina, e armazenamento e gerenciamento dos registros.

O primeiro subsistema é formado pela ESP32-CAM e pelo sensor de imagem OV2640. A câmera permanece integrada à placa e utiliza os recursos internos do ESP32 para aquisição dos quadros.

O segundo subsistema corresponde à detecção da aproximação do animal. O sensor HC-SR04 é responsável pela medição da distância entre o dispositivo e o objeto presente na região monitorada. Na versão 1.0, a leitura do sensor ultrassônico utilizava a função bloqueante pulseIn(), a qual travava a execução do processador enquanto aguardava o término do pulso de retorno. Na versão 2.0, o eco é lido por interrupção de hardware, sem espera bloqueante.

O terceiro subsistema é responsável pela iluminação infravermelha. A leitura do ambiente é realizada pelo módulo LDR com comparador LM393. Quando configurado no modo automático, o firmware utiliza essa informação para determinar se o ambiente está claro ou escuro. Caso seja necessário, os LEDs infravermelhos são acionados através do MOSFET.

O quarto subsistema corresponde ao armazenamento. O cartão MicroSD é utilizado para armazenar as imagens e o arquivo CSV. As fotografias são organizadas em subpastas por data dentro de “/fotos”, enquanto o banco de eventos é mantido nos arquivos “/registros.csv” (por foto) e “/visitas.csv” (por visita).

O fluxo geral inicia-se com o monitoramento do ambiente. O HC-SR04 realiza medições periódicas e o firmware verifica se a distância está dentro do limite configurado. Quando uma aproximação é identificada, são necessárias leituras consecutivas suficientes para validar a presença. Caso a condição seja confirmada e o intervalo mínimo desde a última fotografia tenha sido respeitado, o sistema solicita uma nova captura.

Em condições de baixa luminosidade, o mecanismo de iluminação infravermelha é preparado antes da aquisição da imagem. A câmera realiza a captura, o RTC fornece a data e a hora do evento e o arquivo JPEG é armazenado no cartão MicroSD. Em seguida, uma entrada é acrescentada ao arquivo CSV e o registro passa a fazer parte do histórico disponível na interface do equipamento.

Essa organização estabelece uma cadeia de processamento que pode ser representada como:

Detecção → Confirmação → Preparação da Iluminação → Captura → Registro de Data E Hora → Armazenamento → Registro CSV → Atualização da IHM.

A arquitetura foi concebida para que cada etapa tenha uma responsabilidade definida. Essa separação facilita a manutenção do firmware e permite futuras ampliações, como sincronização via Wi-Fi, processamento de imagens e integração com aplicações externas.

2.4 Resultados Preliminares: Solução Inicial

Nesta seção, serão apresentados os resultados das etapas “ouvir e interpretar o contexto”, “criar/prototipar” e “implementar/testar”. No primeiro caso, foi realizada a consulta com o público-alvo para compreender o contexto do projeto. Na segunda etapa, são apresentados os resultados obtidos relacionados à versão preliminar do protótipo.

2.4.1 Resultados do levantamento com o público-alvo

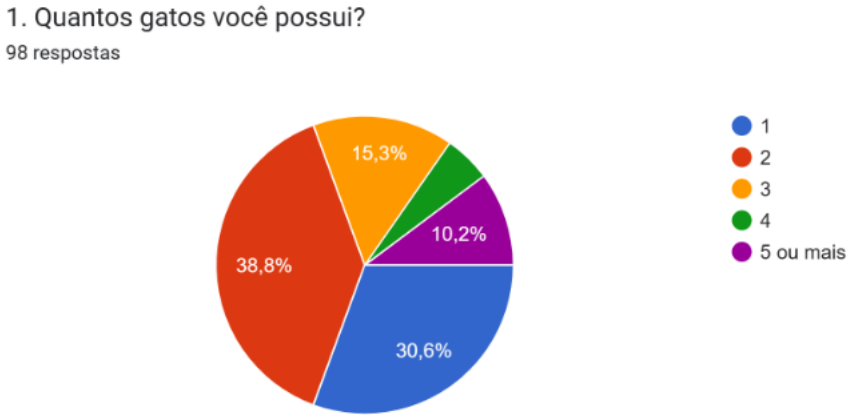
A partir da consulta com o público-alvo através de um formulário online, foram obtidas informações relacionadas ao contexto do projeto. O formulário foi respondido por 98 pessoas

no total, a partir do qual foram obtidas informações relevantes quanto ao perfil do público e suas necessidades quanto ao monitoramento dos hábitos de hidratação dos gatos domésticos e ao interesse quanto às funcionalidades do sistema proposto.

A primeira etapa do questionário tinha como objetivo conhecer o perfil dos tutores e do(s) gato(s). Foram levantadas informações em relação à quantidade de gatos no domicílio, se vivem exclusivamente dentro de casa, idade e idas ao veterinário.

Conforme apresentado na Figura 1, 30,6% do público possui 1 gato. Os dados mostram que a maioria dos tutores possui mais de um gato, o que pode indicar a dificuldade de monitorar a saúde e o hábito de beber água de cada animal.

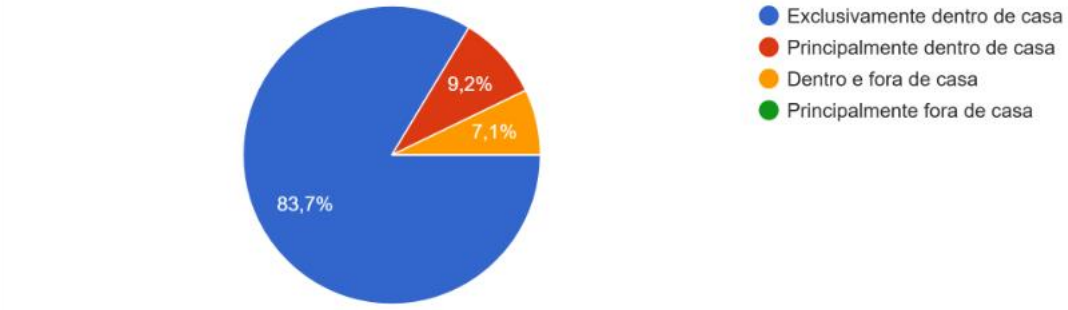
Figura 1 – Quantidade de gatos que os tutores possuem.



Fonte: Elaboração própria dos autores (2026).

Conforme apresentado na Figura 2, a grande maioria dos gatos vive exclusivamente dentro de casa, representando um total de aproximadamente 84%. Esses dados indicam que a grande maioria dos gatos não possui acesso à rua, com uma rotina de maior segurança e controle do bem-estar pelos tutores.

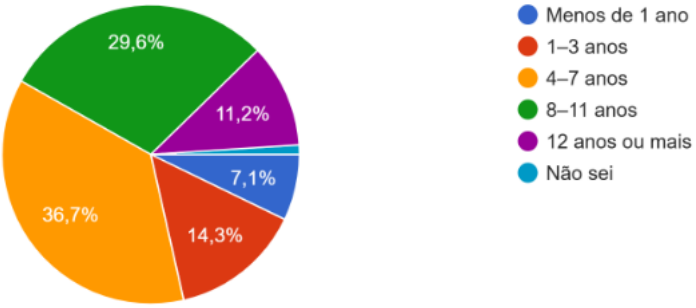
Figura 2 – Dados quanto aos gatos viverem dentro ou fora de casa.



Fonte: Elaboração própria dos autores (2026).

O perfil de idade dos felinos mais velhos dos tutores é representado em maior número por animais jovens, na faixa entre 0 a 7 anos (Figura 3). Porém, uma alta porcentagem dos gatos está na faixa de idade acima dos 8 anos, nos quais estão inclusos os adultos e os idosos, sendo estes últimos os mais acometidos por problemas renais.

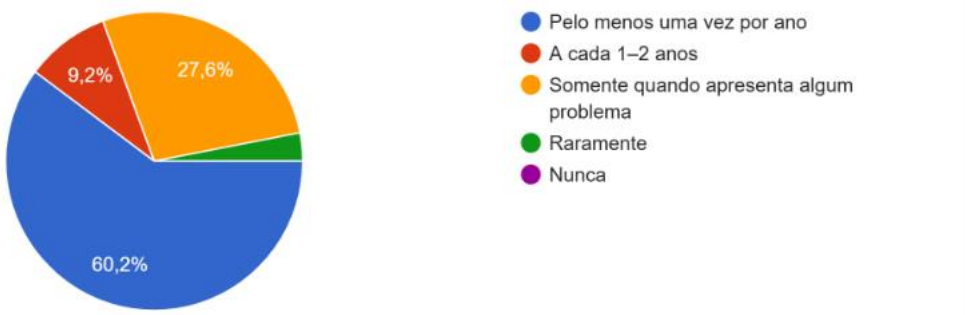
Figura 3 – Faixa etária dos gatos.



Fonte: Elaboração própria dos autores (2026).

Um dado importante é o costume de levar os animais para consultas preventivas no veterinário (Figura 4): em torno de 60% dos entrevistados relataram que as visitas ocorrem, pelo menos, uma vez por ano. Porém, aproximadamente 28% dos tutores costumam levar seus gatos ao veterinário apenas quando apresentam algum problema de saúde. Esses dados indicam a grande preocupação dos tutores com a saúde e a prevenção de doenças nos felinos, já que levam os mesmos para consultas no veterinário com finalidade preventiva pelo menos uma vez por ano.

Figura 4 – Costume de levar os gatos ao veterinário para consultas preventivas.



Fonte: Elaboração própria dos autores (2026).

A segunda etapa do questionário tinha como objetivo o levantamento de dados para compreender o comportamento de ingestão de água pelos felinos domésticos, e se os tutores conseguem observar ou se preocupam com isso.

Conforme apresentado na Figura 5, aproximadamente 41% do público-alvo consegue ter uma estimativa. Por outro lado, mais de 50% dos entrevistados informaram que possuem apenas uma ideia aproximada (30,6%) ou não sabem (20,4%). Os resultados mostram que os tutores tentam quantificar a quantidade de água ingerida pelos gatos, mas que possuem dificuldade para fazer isso.

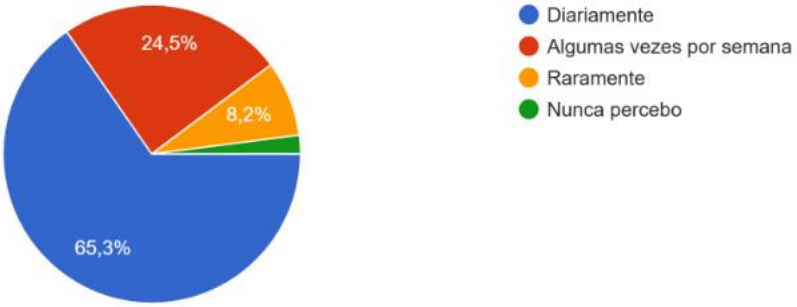
Figura 5 – Informações se os tutores sabem o quanto os gatos bebem de água por dia.



Fonte: Elaboração própria dos autores (2026).

A maioria dos tutores informou que costuma observar os felinos consumindo água (Figura 6). A frequência na qual conseguem realizar essa ação varia entre diariamente (65,3%) e nunca percebo (2%). Os resultados mostram que os tutores fazem o possível para monitorar o comportamento de hidratação dos gatos, a maioria diariamente.

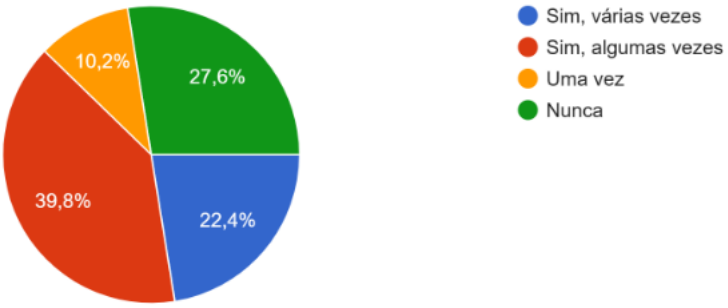
Figura 6 – Costume de observar se o gato bebe água.



Fonte: Elaboração própria dos autores (2026).

Em relação à preocupação quanto à possibilidade de o animal estar bebendo pouca água, conforme mostrado na Figura 7, mais da metade dos cuidadores manifestou que já se preocupou pelo menos uma vez. Por outro lado, em torno de 28% dos entrevistados informaram que nunca se preocuparam. Já na situação inversa, ou seja, quanto à preocupação pela ingestão excessiva de água pelos gatos (Figura 7), quase 75% dos tutores assinalaram que não ficaram preocupados com esse comportamento.

Figura 7 – Nível de preocupação dos tutores quanto ao gato estar bebendo pouca água.

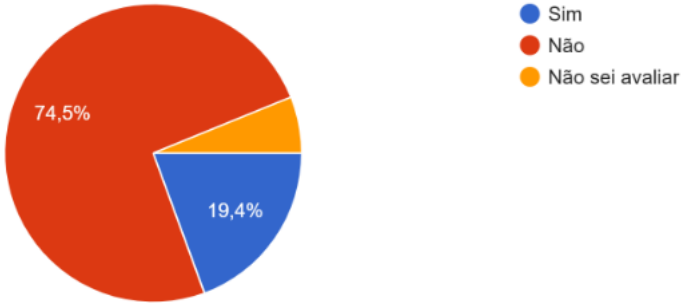


Fonte: Elaboração própria dos autores (2026).

Esses dados em conjunto indicam que os tutores estão observando, dentro do possível, os hábitos dos gatos quanto à água, porém que a maior preocupação é se estão bebendo menos do que deveriam.

Figura 8 – Nível de preocupação dos tutores quanto ao gato estar bebendo muita água

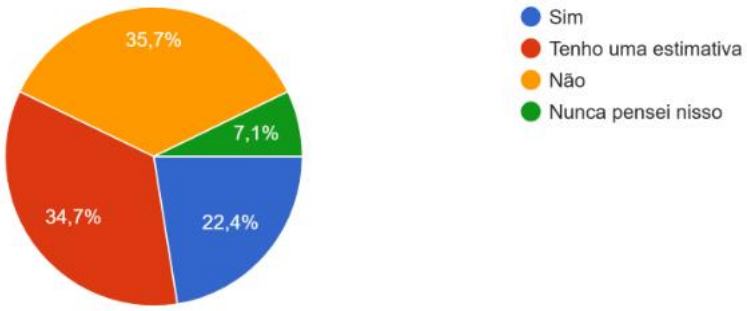
.



Fonte: Elaboração própria dos autores (2026).

Diante do dispositivo em desenvolvimento neste projeto, um dado importante para o levantamento foi em relação ao monitoramento da ingestão de água pelos gatos quando o tutor está fora de casa (Figura 9): 35,7% dos tutores não sabem ou conseguem estimar se esse evento ocorreu durante os períodos de ausência. Os resultados mostram que os tutores estão prestando atenção e buscam acompanhar o comportamento de hidratação dos gatos e o quanto o sistema de monitoramento pode ser útil para auxiliar na demanda.

Figura 9 – Monitoramento dos tutores quanto ao gato estar bebendo água em suas ausências.



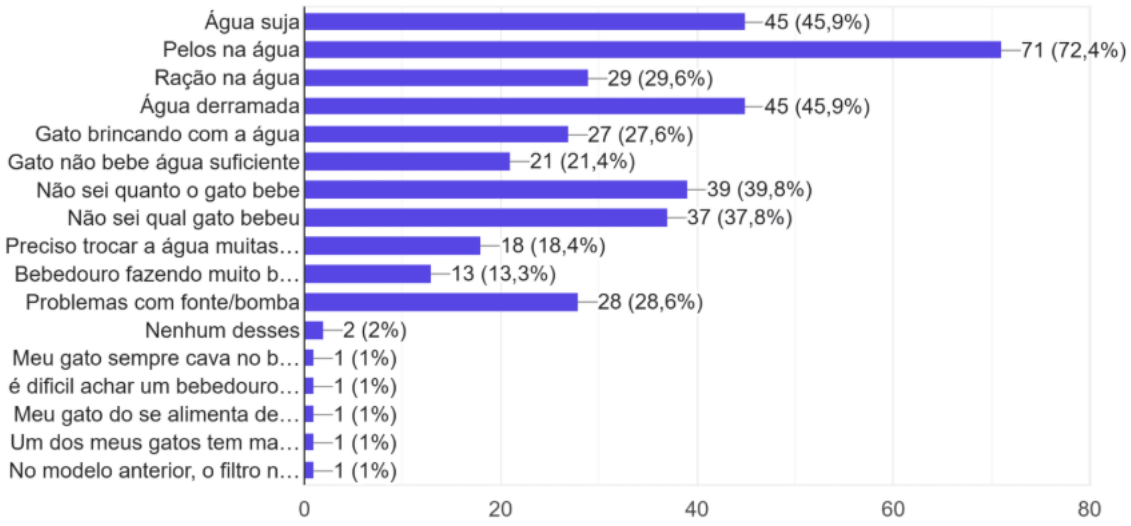
Fonte: Elaboração própria dos autores (2026).

Na terceira etapa do questionário, as perguntas visavam conhecer o levantamento de informações relacionadas aos problemas e necessidades referentes à ingestão de água pelos gatos.

Conforme apresentado na Figura 10, foram obtidos dados sobre os problemas relacionados à água ou ao bebedouro do gato. Dentre os principais problemas apontados, podem

ser destacados a presença de pelos, água suja, água derramada, e não saber identificar se o gato bebeu água ou qual gato realizou a ação. Os dados mostram que as maiores preocupações dos entrevistados residem na qualidade da água que está sendo oferecida aos gatos, a sua disponibilidade e se os animais estão bebendo.

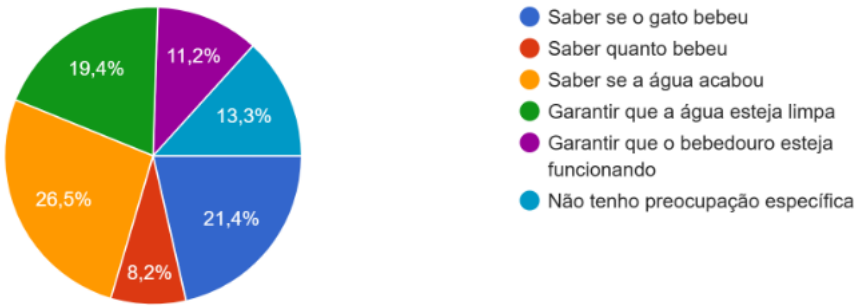
Figura 10 – Problemas apontados em relação à água disponível para os felinos.



Fonte: Elaboração própria dos autores (2026).

Outros pontos importantes em relação ao tempo do tutor fora de casa, com grande relação ao dispositivo em desenvolvimento neste projeto, são os dados quanto à maior preocupação que os tutores têm em relação à água (Figura 11): se foi ingerida, o quanto foi ingerida, se tem água ou se esta está limpa e, no caso de bebedouros automatizados, se estes estão funcionando. A grande maioria dos tutores apresenta alguma dessas preocupações, com apenas 13,3% não tendo nenhuma em específico. Em torno de 21,4% dos tutores se preocupam em saber se o gato bebeu e 8,2% em saber o quanto foi ingerido. Esses dados apontam que os tutores se preocupam em saber se os gatos sob sua tutela estão se hidratando adequadamente, o que pode ser facilitado por um sistema de monitoramento quando estão fora de casa.

Figura 11 – Maior preocupação dos tutores quanto à água quando estão fora de casa.

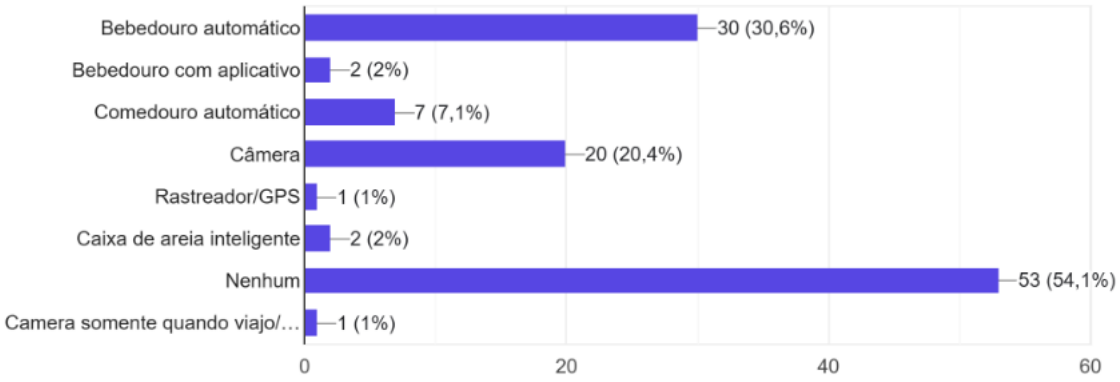


Fonte: Elaboração própria dos autores (2026).

Na quarta etapa do questionário, as questões visavam obter informações quanto ao uso de tecnologias para o cuidado dos gatos, relacionadas à ingestão de água, alimentação, monitoramento e localização.

A primeira questão dessa etapa está relacionada ao fato de os cuidadores possuírem algum dispositivo inteligente para o cuidado do animal (Figura 12). Os dados mostram que 53 tutores não possuem um sistema com essa finalidade. Por outro lado, 20 tutores têm câmeras para o monitoramento dos animais. Quanto à hidratação dos gatos, 30 cuidadores possuem bebedouro automático, indicando uma grande preocupação com a hidratação do pet.

Figura 12 – Utilização de dispositivo inteligente para o gato.

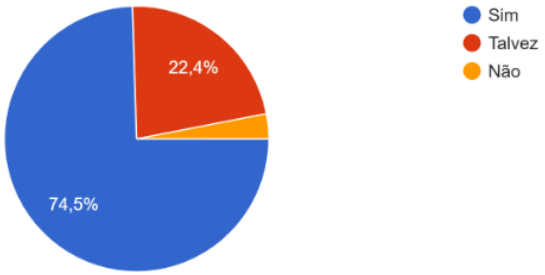


Fonte: Elaboração própria dos autores (2026).

Os tutores foram questionados quanto ao interesse em receber informações pelo celular sobre o consumo de água pelos felinos (Figura 13). Os resultados apontam que a maioria dos cuidadores (74,5%) gostaria de ser informada sobre a hidratação dos gatos através do celular.

Esse requisito para o dispositivo em desenvolvimento pode ser adicionado futuramente, após a validação do protótipo final.

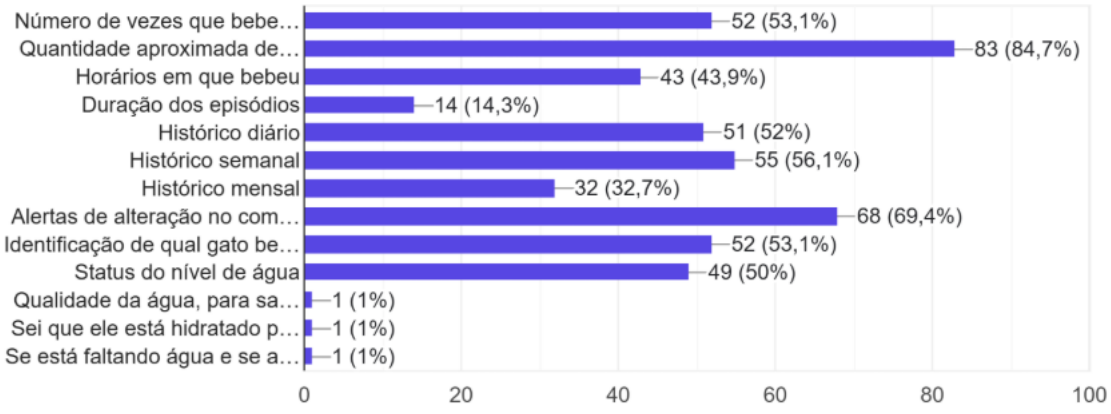
Figura 13 – Interesse do público-alvo em receber informações sobre o consumo de água pelos gatos.



Fonte: Elaboração própria dos autores (2026).

Dentre as informações relacionadas à hidratação a serem recebidas pelo celular, os tutores foram questionados sobre quais seriam as mais úteis para eles. Conforme apresentado na Figura 14, a maior parte dos tutores – 83 deles – gostaria de saber a quantidade aproximada de quanto o gato bebeu. Outros dados considerados importantes para os tutores são em relação ao número de vezes que bebeu água (52 deles), alertas de alteração no comportamento (68 tutores), identificação de qual gato bebeu (52 tutores) e, principalmente, um histórico dos eventos. Estes dados corroboram a escolha dos parâmetros escolhidos para serem implementados e melhorados no protótipo, visando atender o público-alvo.

Figura 14 – Informações que os cuidadores consideram mais úteis quanto à hidratação dos gatos.

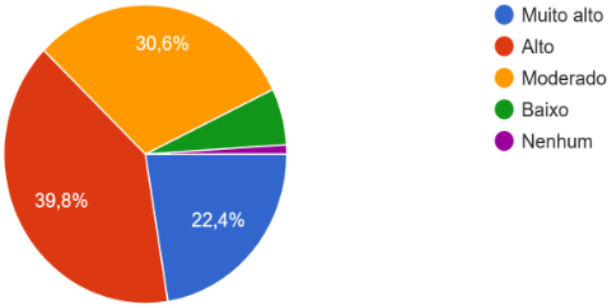


Fonte: Elaboração própria dos autores (2026).

Na quinta fase, as perguntas visavam avaliar o conceito do produto para os tutores, ou seja, o interesse, as funcionalidades mais importantes e se consideram útil o monitoramento com câmera integrada.

Conforme apresentado na Figura 15, na qual os tutores foram questionados quanto ao interesse em adquirir o dispositivo proposto, mais da metade possui um alto ou muito alto interesse em adquirir, e aproximadamente 31% deles possuem interesse moderado. Estes dados validam a proposta do dispositivo e sua possível viabilidade comercial.

Figura 15 – Nível de interesse dos cuidadores em utilizar o sistema conforme proposto no projeto.

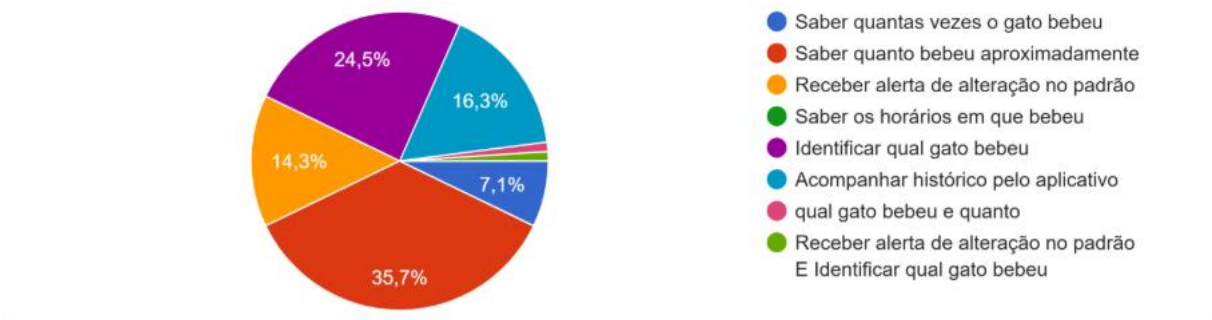


Fonte: Elaboração própria dos autores (2026).

Ainda em relação ao dispositivo em desenvolvimento neste projeto, o público-alvo foi questionado sobre qual funcionalidade seria mais importante (Figura 16). A maioria dos tutores – aproximadamente 36% – gostaria de saber o quanto o gato bebeu aproximadamente, assim como identificar qual deles bebeu (~25%). Outro dado importante foi o interesse em acompanhar o histórico por um aplicativo, considerada importante para 16,3% dos entrevistados. Esses dados indicam quais são os requisitos mais importantes para os entrevistados sobre qual é a prioridade de informação no caso de o produto em desenvolvimento ter um aplicativo para tal acompanhamento. No caso do dispositivo proposto, algumas dessas informações poderão ser obtidas no display, já que primeiramente se trata de um protótipo para funcionamento autônomo.

Figura 16 – Funcionalidade no sistema que os tutores consideram ser mais importante

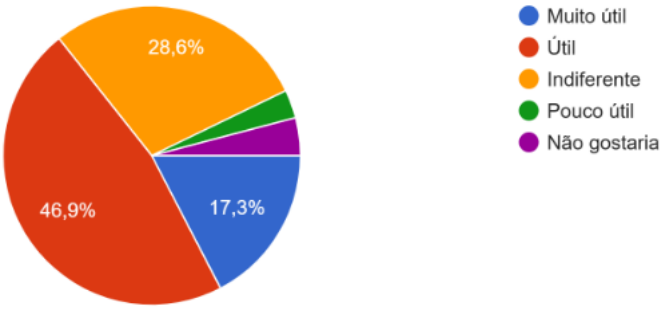
.



Fonte: Elaboração própria dos autores (2026).

Em relação à utilidade de uma câmera integrada ao monitor de hidratação felina, conforme apresentado na Figura 17, mais da metade do público reconhece ser útil para essa finalidade, indicando que essa melhoria é muito importante de ser realizada no protótipo deste projeto.

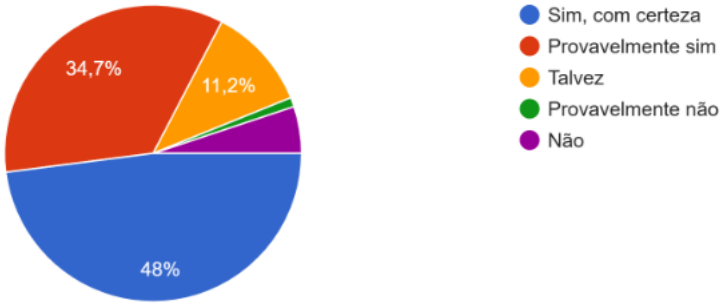
Figura 17 – Utilidade de câmera de monitoramento integrada ao dispositivo segundo os tutores.



Fonte: Elaboração própria dos autores (2026).

Diante do fato de o sistema proposto neste projeto ser para instalação no bebedouro atual dos gatos, sem necessidade de substituí-lo, ou seja, não necessitar de um gasto maior por parte dos tutores, estes foram questionados quanto à possibilidade de utilizar o mesmo (Figura 18). Neste cenário, 48% responderam "Sim, com certeza" e aproximadamente 35% "Provavelmente sim", indicando a grande viabilidade comercial do sistema proposto.

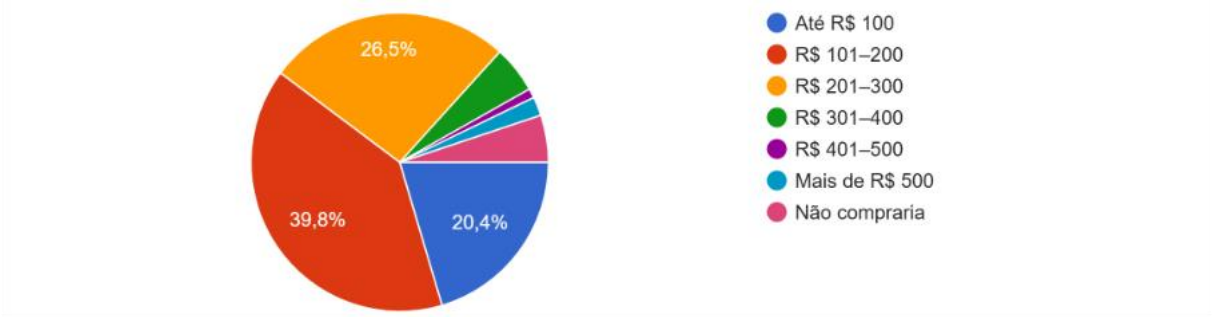
Figura 18 – Possibilidade de adquirir um dispositivo que possa ser acoplado ao bebedouro existente segundo os cuidadores.



Fonte: Elaboração própria dos autores (2026).

A parte financeira costuma ser um dos requisitos para a aquisição de itens para o cuidado dos animais domésticos. Neste sentido, os entrevistados foram questionados sobre qual faixa de valor consideram aceitável pagar pelo dispositivo (Figura 19). A maioria dos tutores – 39,8% – informou que está disposto a pagar entre R$ 100,00 e R$ 200,00. Uma parte deles acredita ser aceitável pagar um pouco acima desse valor (26,5%), na faixa entre R$ 200,00 e R$ 300,00. Esses dados apontam que os tutores possuem interesse no dispositivo, mas que este não possua custo elevado.

Figura 19 – Valor considerado aceitável pelos entrevistados para pagar no dispositivo.

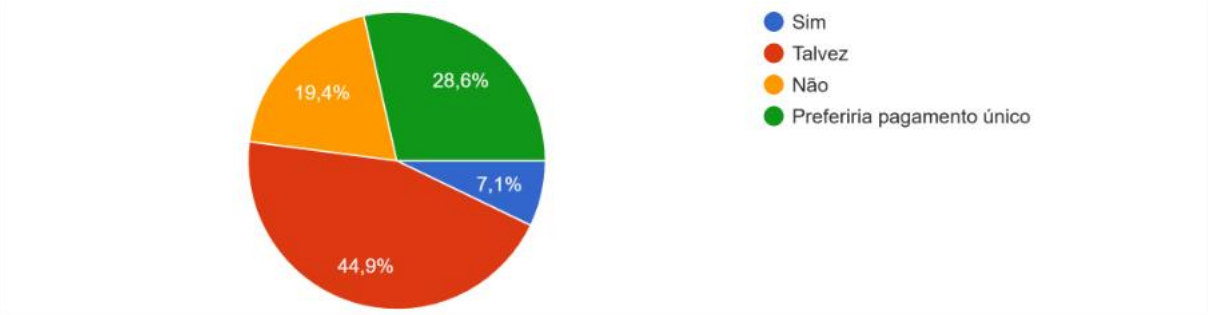


Fonte: Elaboração própria dos autores (2026).

Visando desenvolver um plano de negócios para o sistema, o público foi questionado quanto à possibilidade de pagar uma mensalidade para ter armazenamento do histórico e recursos adicionais. Conforme apresentado na Figura 20, 44,9% do público considerou essa possibilidade viável, em contraste com 28,6% que apontaram a preferência por pagamento único. Os dados mostram que, caso o plano de negócios envolva uma assinatura para o

acompanhamento de um histórico e recursos adicionais, é possível que o público aceite pagar por isso, visando um maior cuidado com o animal.

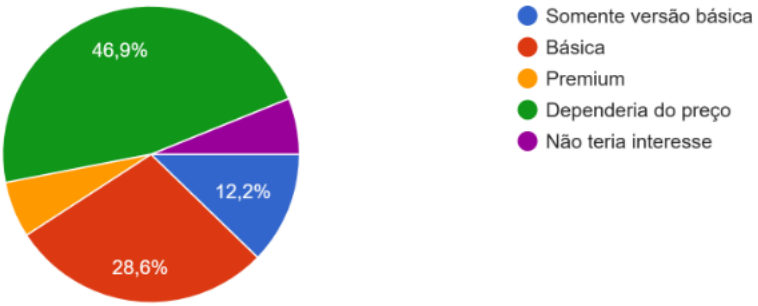
Figura 20 – Possibilidade de pagar mensalidade para o serviço de monitoramento segundo os tutores.



Fonte: Elaboração própria dos autores (2026).

Ainda neste cenário, caso houvesse uma versão básica sem mensalidades e uma versão premium com recursos adicionais (Figura 21), para aproximadamente 47% dos entrevistados a preferência seria determinada a depender do valor. Os dados indicam a viabilidade de um plano de negócios em que exista uma assinatura mensal premium com mais recursos, desde que o valor seja acessível para o público.

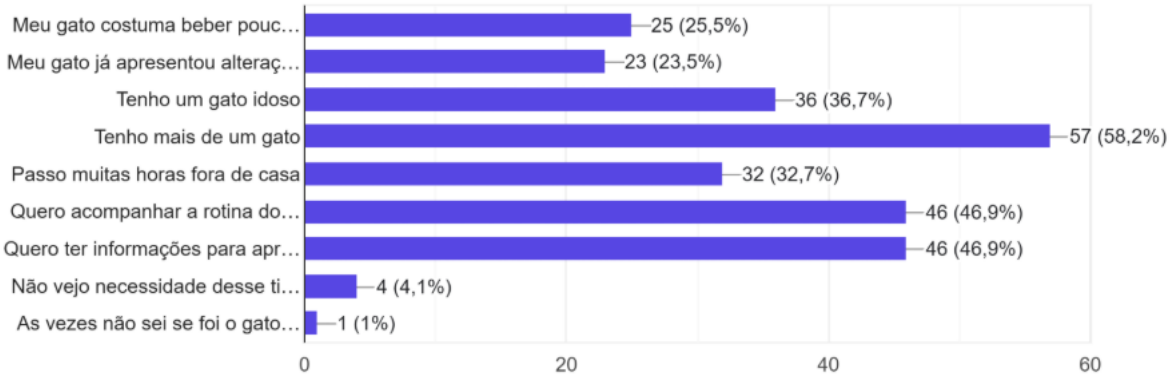
Figura 21 – Preferência dos tutores em relação às versões básica sem mensalidade ou premium com assinatura mensal para o serviço de monitoramento.



Fonte: Elaboração própria dos autores (2026).

Conforme apresentado na Figura 22, foram levantados dados quanto às situações que fariam o tutor considerar o sistema de monitoramento de hidratação felina útil. Os dados mostram que muitos tutores veem a utilidade pelo fato de possuírem mais de um gato, quererem acompanhar a rotina e apresentar dados para o veterinário. Além disso, 36 deles possuem um gato idoso e 23 passam muito tempo fora de casa.

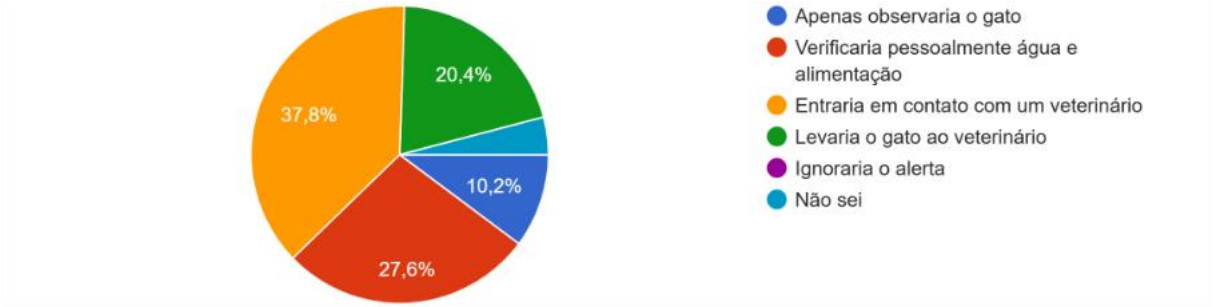
Figura 22 – Situações consideradas para avaliar a utilidade de um sistema de monitoramento de água segundo os tutores.



Fonte: Elaboração própria dos autores (2026).

Por último, conforme apresentado na Figura 23, foram obtidos dados em relação ao possível alerta da aplicação quanto à mudança no padrão de consumo de água pelo gato. Os resultados apontam que a maioria dos tutores (~38%) utilizaria a informação para entrar em contato com o veterinário e avaliar a saúde do felino, e 20,4% deles levariam o gato para atendimento. Esses dados mostram a importância de um dispositivo de monitoramento que consiga armazenar o histórico dos hábitos e informar mudanças nesse padrão de comportamento visando a saúde e o bem-estar dos animais.

Figura 23 – Dados em relação ao possível alerta da aplicação quanto à mudança no padrão de consumo de água pelo gato e atitudes resultantes dos tutores.



Fonte: Elaboração própria dos autores (2026).

Os dados da pesquisa, em conjunto, mostram a importância dos gatos para os tutores e o quanto eles se preocupam com a saúde e o bem-estar de seus animais. Muitos tutores possuem mais de um gato, sendo em sua maioria adultos ou idosos. Esses fatores mostram a necessidade de um sistema de monitoramento do comportamento dos gatos quanto à água, uma vez que são residências em que é mais difícil saber se cada gato está bebendo a quantidade adequada,

principalmente quando se leva em consideração os idosos, os quais são os que possuem maior prevalência de doenças relacionadas aos rins.

Além disso, os resultados demonstram que os cuidadores gostariam de adquirir um sistema que possibilitasse o monitoramento da hidratação dos gatos, principalmente visando acompanhar esses eventos e, caso informados de mudanças nesse padrão, possam acionar o médico veterinário. Os dados considerados importantes pelos tutores estão relacionados principalmente à saúde desses animais, o que os fez considerar pagar uma assinatura caso o sistema proposto tenha essa opção, desde que os valores sejam acessíveis e permitam acompanhar o histórico do gato. Neste sentido, a pesquisa indica o forte potencial comercial do sistema em desenvolvimento.

2.5 Resultados Preliminares da Solução Inicial

A presente seção apresenta a avaliação conceitual e a análise crítica dos resultados obtidos por meio do desenvolvimento do firmware, da modelagem lógica e das simulações conceituais do protótipo HydraPet Vision em sua versão 2.0. Reforça-se que, por se tratar de uma prova de conceito avançada e de um projeto voltado ao desenvolvimento de firmware e validação lógica, o dispositivo físico não foi montado em bancada. O trabalho fundamenta-se na especificação de arquitetura, na codificação em linguagem C++ e na verificação analítica, de modo que todas as avaliações decorrem de rastreamento lógico de cenários, cálculos de engenharia elétrica e simulação conceitual do comportamento do sistema.

Essa escolha metodológica e arquitetural assegura que o dispositivo seja projetado para operar de maneira autônoma e prioritariamente *offline*, sem depender de redes locais, latências ou quedas de conexão, e preservando a segurança e a privacidade dos dados no ambiente doméstico.

Cabe ressaltar, como condicionante prática no desenvolvimento de sistemas embarcados com câmera, que plataformas de simulação virtual como Tinkercad Circuits, Wokwi ou Proteus Design Suite não oferecem suporte à emulação do periférico de câmera OV2640, da controladora de DMA da câmera do ESP32, nem do barramento avançado de memória PSRAM/SDMMC associado à placa AI-Thinker ESP32-CAM. Por essa razão, a verificação do sistema apoia-se na análise do código-fonte e na verificação estática das rotinas cooperativas implementadas.

A arquitetura lógica estabeleceu os princípios fundamentais de funcionamento do sistema, utilizando a ESP32-CAM como unidade central de processamento, o sensor

ultrassônico HC-SR04 para identificação da aproximação do animal, o módulo RTC DS1307 para registro temporal dos eventos, o display TFT para apresentação das informações e o cartão MicroSD para armazenamento das imagens. Para contornar a limitação de GPIOs da ESP32-CAM, visto que a câmera OV2640 e o leitor MicroSD ocupam diversas linhas internas, especificou-se a integração do expansor de entradas e saídas PCF8574 ao barramento I2C compartilhado. Esse componente concentra as entradas dos três botões de navegação, a saída de acionamento do sensor ultrassônico, o comando da iluminação infravermelha e a leitura do módulo LDR responsável pela identificação das condições de luminosidade ambiente, ampliando as funções disponíveis sem comprometer os recursos internos.

No tocante ao sensoriamento de distância, a leitura do sensor ultrassônico foi projetada com uma rotina baseada em interrupção no pino ECHO (GPIO 3), eliminando funções bloqueantes e permitindo que o ciclo principal continue executando as demais tarefas do sistema. Adicionalmente, o firmware incorpora um mecanismo de confirmação de presença por leituras consecutivas configuráveis, reduzindo a probabilidade de disparos indevidos provocados por ruídos ou reflexões ocasionais.

O sistema de iluminação foi especificado com a incorporação de seis LEDs infravermelhos de 850 nm, acionados por um estágio de chaveamento com MOSFET e controlados via PCF8574. A operação automática depende da leitura estabilizada do sensor LDR, permitindo que a iluminação seja ativada em ambientes de baixa luminosidade, com opções de modo automático, permanentemente ligado ou desligado.

No âmbito do software, a utilização da memória não volátil NVS (por meio da biblioteca *Preferences*) garante a preservação dos parâmetros de configuração, enquanto a organização dos dados foi aprimorada com a gravação de fotografias em formato JPEG no diretório /fotos e a manutenção do banco tabular /registros.csv com informações contextuais de data, hora, distância e estado do infravermelho. Desse modo, o projeto propõe uma plataforma embarcada com arquitetura modular, preparada para futuras montagens físicas e expansões no setor de tecnologia voltado ao bem-estar animal (*pet tech*).

2.6 Trabalhos Correlatos e Comparativo Tecnológico

O desenvolvimento de sistemas embarcados voltados ao bem-estar animal (*pet tech*) tem recebido atenção crescente na literatura acadêmica e no mercado de tecnologia, impulsionado pela popularização de microcontroladores de baixo custo com conectividade e recursos multimídia. No entanto, parte dos produtos comerciais consultados limita-se a fontes de água

dinâmicas desprovidas de telemetria ou sistemas complexos baseados exclusivamente em infraestrutura de rede em nuvem, o que implica vulnerabilidade a quedas de conexão e dependência de servidores externos.

Para posicionar tecnologicamente o HydraPet Vision 2.0, realizou-se, em outubro de 2026, um levantamento não sistemático em bases abertas (repositórios institucionais, periódicos de acesso aberto e páginas oficiais de fabricantes), com termos relacionados a bebedouro e comedouro inteligente, ESP32-CAM e monitoramento de gatos. Foram selecionados cinco trabalhos consultados na íntegra ou em página oficial: quatro projetos acadêmicos (FITRIANI, 2024; CASTILLO-ARCEO; RENTERIA-FLORES; SANTANA-MANCILLA, 2024; UNTIVEROS-PEÑALOZA; CASTRO-VARGAS, 2025; KOTWAL et al., 2025) e um produto comercial com câmera (PETKIT, [s.d.]). A Tabela 1 resume o que cada fonte informa; onde a fonte não informa um item, isso é indicado, e a ausência de informação não deve ser lida como ausência do recurso.

Tabela 1 – Comparativo entre o HydraPet Vision 2.0 e projetos correlatos desenvolvidos com arquiteturas similares.

Projeto / Autor (Ano)	Plataforma e Sensores	Recursos Visuais	Armazenamento e Interface	Autonomia / Conectividade
Monitoramento e controle de alimentação de gatos com HC-SR04 e ESP32-CAM (FITRIANI, 2024)	ESP32-CAM + HC-SR04	Câmera ESP32-CAM; registro de fotos não informado	Armazenamento não informado / controle por Telegram	Depende de rede (Telegram)
Smart pet feeder com IoT e aprendizado profundo (CASTILLO-ARCEO; RENTERIA-FLORES; SANTANA-MANCILLA, 2024)	Arduino Mega 2560 + HX711 + HC-SR04	Câmera com YOLOv5; registro de fotos não informado	Armazenamento não informado / arquitetura IoT	Conectividade IoT (detalhes não informados)
Alimentador com configuração dinâmica de rede, ESP32 e Blynk 2.0	ESP32 + HC-SR04 (nível de água)	Não informada	Registro de eventos no firmware e no Blynk / aplicativo	Depende de Wi-Fi (Blynk 2.0)

(UNTIVEROS-PEÑALOZA; CASTRO-VARGAS, 2025)				
Deep-feed: alimentador inteligente com aprendizado profundo (KOTWAL et al., 2025)	ESP8266 + sensores de proximidade, ultrassônico e de peso	Câmera com CNN; armazenamento das imagens não informado	Armazenamento não informado / aplicativo Blynk	Depende de Wi-Fi (Blynk)
Fonte EverSweet Ultra com câmera, produto comercial (PETKIT, [s.d.])	Não informada	Câmera com lente de 140° e reconhecimento de até 15 animais	Histórico por visita / aplicativo	Wi-Fi e Bluetooth; nuvem não especificada
HydraPet Vision 2.0 (Presente Trabalho, 2026)	ESP32-CAM + PCF8574 + HC-SR04 + LDR	OV2640 + Visão Noturna IR (850 nm)	MicroSD (CSV/JPEG) / Display TFT 2,4"	Offline: sem uso de rede na versão atual

Fonte: Elaborado pelos autores (2026).

Observa-se, na Tabela 1, que os projetos acadêmicos levantados tratam principalmente de comedouros, e não de bebedouros. Fitriani (2024) e Castillo-Arceo, Renteria-Flores e Santana-Mancilla (2024) combinam o sensor ultrassônico HC-SR04 com câmera (ESP32-CAM e uma câmera com YOLOv5, respectivamente); Untiveros-Peñaloza e Castro-Vargas (2025) usam o ESP32 e o HC-SR04 para medir o nível de água e dependem de Wi-Fi e do aplicativo Blynk; Kotwal et al. (2025) empregam uma câmera com redes neurais convolucionais em um alimentador conectado ao Blynk. Nas fontes consultadas, não consta o armazenamento local de fotografias em cartão de memória como mecanismo principal de registro.

No segmento comercial, a fonte PETKIT EverSweet Ultra com câmera (PETKIT, [s.d.]) oferece câmera com reconhecimento de animais, histórico por visita e conectividade Wi-Fi e Bluetooth por aplicativo, a um preço informado de US$ 249,99 na página do fabricante; a Petlibro Dockstream 2 (PETLIBRO, [s.d.]) e a Sure Petcare Felaqua Connect (SURE PETCARE, [s.d.]) registram o consumo e as sessões de bebida por aplicativo, sem câmera informada. No campo veterinário, Nicolae e Codreanu (2024) estudaram alterações do comportamento de beber de gatas gestantes por meio de gravações de uma câmera de segurança com visão noturna, o que ilustra o interesse no registro audiovisual do ato de beber, ainda que sem um sistema embarcado dedicado.

O HydraPet Vision 2.0 busca combinar, em uma única plataforma de custo estimado de R$ 285,40:

1. **Autonomia operacional integral (offline):** Armazenamento estruturado local em formato tabular CSV e banco de imagens JPEG em cartão MicroSD, sem exigência de infraestrutura de rede externa.
2. **Interface Homem-Máquina local completa:** Display TFT de 2,4 polegadas operando com galeria fotográfica integrada, eliminando a dependência de aplicativos de terceiros para a auditoria imediata pelo tutor.
3. **Gerenciamento avançado de restrições de hardware:** Utilização do expansor I/O PCF8574 para preservação das GPIOs da ESP32-CAM e adoção de sensoriamento de presença não bloqueante orientado por interrupções de hardware.

Portanto, a partir dos trabalhos consultados, o HydraPet Vision 2.0 posiciona-se como um registrador fotográfico local de visitas ao bebedouro, de custo estimado reduzido (Seção 4.9), que não depende de rede, de nuvem nem de aplicativo. Trata-se de uma escolha com contrapartidas: o sistema não mede o volume ingerido, não identifica o animal e não oferece acesso remoto, funções presentes em parte dos produtos comerciais. O levantamento é não sistemático e não permite afirmar que a combinação proposta seja inédita.

2.7 A Natureza IoT do Sistema Frente à Operação Autônoma e Offline

Embora a arquitetura clássica de sistemas de Internet das Coisas (IoT) seja frequentemente associada à transmissão contínua de dados para servidores em nuvem, a evolução moderna dos sistemas embarcados expandiu esse conceito para contemplar a computação de borda (*Edge Computing*) e nós autônomos de processamento local (BAHGA; MADISETTI, 2014; LEE; SESHIA, 2017). Na definição fundamentada por Bahga e Madisetti (2014), um sistema de IoT é composto por blocos funcionais que envolvem sensoriamento, atuação, processamento local e comunicação, podendo operar de maneira autônoma ou integrada a redes externas.

Sob essa perspectiva conceitual, o protótipo HydraPet Vision 2.0 pode ser enquadrado como nó de borda de IoT, ainda que sem conectividade na versão atual, mesmo adotando uma estratégia de operação prioritariamente offline. A escolha arquitetural de processar e armazenar os dados localmente utilizando um banco de dados tabular em formato CSV e persistência de

imagens JPEG em cartão MicroSD gerenciado pelo SoC ESP32-CAM, configura uma arquitetura de nós autônomos de borda (*Store-and-Forward*). Essa abordagem reduz a dependência de redes locais, preserva a privacidade do ambiente residencial e assegura a integridade dos registros frente a falhas de conectividade (BAHGA; MADISETTI, 2014).

Ademais, a infraestrutura de hardware e software do dispositivo foi concebida sob bases estritamente modulares. O microcontrolador utilizado (ESP32-CAM) possui nativamente recursos de conectividade sem fio (Wi-Fi e Bluetooth), o que significa que a operação offline atual representa uma decisão de projeto voltada à robustez e à segurança local, mantendo a arquitetura integralmente preparada para futuras expansões e integração com plataformas de nuvem em versão subsequente, sem a necessidade de reestruturação do circuito físico ou do firmware básico (LEE; SESHIA, 2017).

3 DESENVOLVIMENTO DO PROTÓTIPO

O presente capítulo descreve o desenvolvimento técnico, a estrutura de hardware, a arquitetura de software e o funcionamento integrado da versão 2.0 do protótipo HydraPet Vision. A elaboração desta versão representa uma evolução em relação à prova de conceito apresentada anteriormente, principalmente em função da necessidade de solucionar limitações relacionadas à disponibilidade de portas de entrada e saída da ESP32-CAM, à confiabilidade do sensoriamento, à operação em ambientes com baixa luminosidade, à organização dos registros e à interação do usuário com o equipamento.

A primeira versão do protótipo estabeleceu os princípios fundamentais de funcionamento do sistema, utilizando a ESP32-CAM como unidade central de processamento, o sensor ultrassônico HC-SR04 para identificação da aproximação do animal, o módulo RTC DS1307 para registro temporal dos eventos, o display TFT para apresentação das informações e o cartão MicroSD para armazenamento das imagens. Entretanto, a evolução do projeto demonstrou que a ampliação das funcionalidades exigia uma reorganização da arquitetura eletrônica e do firmware. A quantidade limitada de GPIOs disponíveis na ESP32-CAM, associada à utilização simultânea da câmera OV2640 e do cartão MicroSD, tornou necessária a adoção de um expansor de entradas e saídas, além de uma estratégia mais estruturada de compartilhamento dos barramentos de comunicação.

Na versão 2.0, essa necessidade foi atendida pela integração do expansor de GPIO PCF8574 ao barramento I2C. O componente passou a concentrar as entradas dos três botões de navegação, a saída de acionamento do sensor ultrassônico, o comando da iluminação infravermelha e a leitura do módulo LDR responsável pela identificação das condições de luminosidade ambiente. Essa alteração permitiu ampliar as funções disponíveis sem comprometer os recursos internos utilizados pela câmera e pelo armazenamento.

Outra evolução importante ocorreu no tratamento do sensor ultrassônico. Na versão anterior, a leitura da distância era realizada mediante chamada bloqueante, enquanto o firmware 2.0 utiliza uma rotina baseada em interrupção no pino ECHO. Dessa forma, o tempo de retorno do sinal ultrassônico pode ser registrado por eventos de subida e descida do sinal digital, reduzindo a dependência de funções bloqueantes e permitindo que o ciclo principal continue executando as demais tarefas do sistema.

A versão atual também incorporou um mecanismo de confirmação de presença. A simples ocorrência de uma leitura dentro da distância configurada não é suficiente para disparar imediatamente a câmera. O firmware exige uma quantidade configurável de leituras

consecutivas dentro da faixa de detecção. Essa estratégia reduz a possibilidade de uma leitura isolada, provocada por ruído ou reflexão ocasional, gerar uma fotografia indevida.

O sistema de iluminação também foi reformulado. A versão 2.0 incorpora seis LEDs infravermelhos com comprimento de onda nominal de 850 nm, acionados por meio de um estágio de chaveamento com MOSFET e controlados pelo PCF8574. A operação automática depende da leitura do sensor LDR, permitindo que a iluminação seja utilizada quando o ambiente apresenta baixa luminosidade. Além disso, o firmware possibilita selecionar entre os modos automático, permanentemente ligado e desligado.

No âmbito do software, a evolução foi ainda mais significativa. O firmware passou a utilizar memória não volátil NVS, por meio da biblioteca Preferences, para preservar as configurações de operação. Assim, parâmetros como distância de detecção, intervalo mínimo entre fotografias, modo de iluminação infravermelha, polaridade do LDR e quantidade de leituras necessárias para confirmação de presença permanecem armazenados mesmo após o desligamento do equipamento.

A organização dos dados também foi aprimorada. Além das fotografias armazenadas em formato JPEG, o sistema mantém um arquivo CSV denominado “registros.csv”, contendo a data, o horário, o caminho da fotografia, a distância medida e a condição da iluminação infravermelha no momento do evento. Essa estrutura transforma cada fotografia em um registro associado a informações contextuais, permitindo posteriormente a análise dos eventos de aproximação.

Dessa forma, a versão 2.0 deixa de ser apenas um dispositivo capaz de detectar aproximações e tirar fotografias e passa a constituir uma plataforma embarcada de registro, consulta e gerenciamento local de eventos relacionados à hidratação felina.

3.1 Arquitetura Geral do Protótipo

A arquitetura do HydraPet Vision 2.0 foi concebida de forma modular, tendo a ESP32-CAM AI-Thinker como unidade central de processamento. A placa concentra o processamento do firmware, o controle da câmera OV2640, o acesso ao cartão MicroSD, a comunicação com o display TFT e a interação com os periféricos externos.

A arquitetura pode ser dividida em cinco subsistemas principais: processamento e captura de imagem; sensoriamento de presença; controle de iluminação; interface homem-máquina; e armazenamento e gerenciamento dos registros.

O primeiro subsistema é formado pela ESP32-CAM e pelo sensor de imagem OV2640. A câmera permanece integrada à placa e utiliza os recursos internos do ESP32 para aquisição dos quadros. A versão 2.0 trabalha com imagens em formato JPEG e resolução QVGA, correspondente a 320 × 240 pixels. Essa resolução foi escolhida de forma coerente com a resolução utilizada pelo display TFT, possibilitando a apresentação das fotografias capturadas na própria interface do dispositivo.

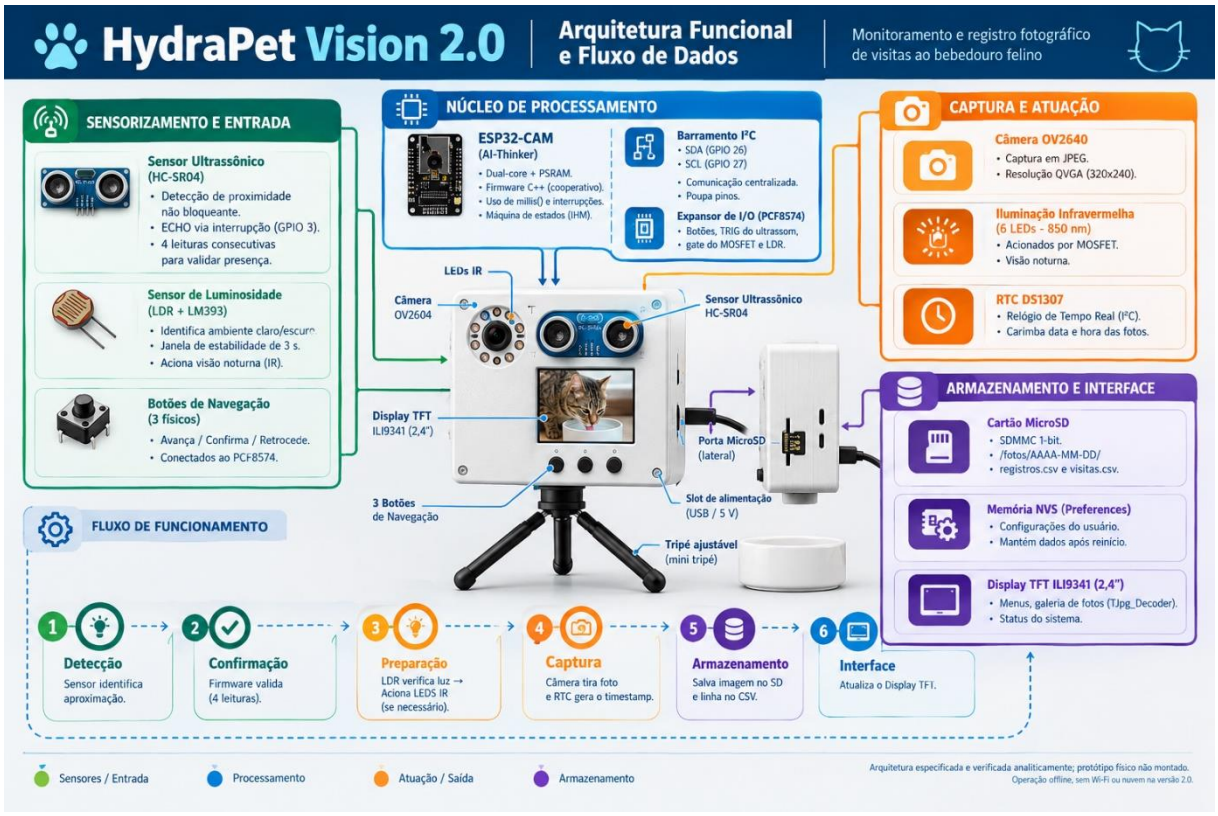
O segundo subsistema corresponde à detecção da aproximação do animal. O sensor HC-SR04 é responsável pela medição da distância entre o dispositivo e o objeto presente na região monitorada. O acionamento do sensor é realizado por uma saída do PCF8574, enquanto o sinal ECHO é conectado diretamente a uma GPIO da ESP32-CAM, utilizando divisor resistivo para adequação do nível lógico.

O terceiro subsistema é responsável pela iluminação infravermelha. A leitura do ambiente é realizada pelo módulo LDR com comparador LM393, conectado a uma entrada do PCF8574. Quando configurado no modo automático, o firmware utiliza essa informação para determinar se o ambiente está claro ou escuro. Caso seja necessário, os LEDs infravermelhos são acionados através do MOSFET.

O quarto subsistema é a IHM, constituída pelo display TFT ILI9341 de 2,4 polegadas e três botões físicos. A interface permite consultar os registros fotográficos, visualizar imagens, modificar a data e a hora do RTC, ajustar parâmetros de detecção e iluminação, restaurar configurações e verificar informações de diagnóstico do sistema.

O quinto subsistema corresponde ao armazenamento. O cartão MicroSD é utilizado para armazenar as imagens e o arquivo CSV. As fotografias são organizadas em subpastas por data dentro de “/fotos”, enquanto o banco de eventos é mantido nos arquivos “/registros.csv”. A Figura 24 apresenta a arquitetura funcional da versão 2.0.

Figura 24 – Arquitetura funcional do protótipo HydraPet Vision 2.0.



Fonte: Elaborado pelos autores (2026).

O fluxo geral inicia-se com o monitoramento do ambiente. O HC-SR04 realiza medições periódicas e o firmware verifica se a distância está dentro do limite configurado. Quando uma aproximação é identificada, são necessárias leituras consecutivas suficientes para validar a presença. Caso a condição seja confirmada e o intervalo mínimo desde a última fotografia tenha sido respeitado, o sistema solicita uma nova captura.

Em condições de baixa luminosidade, o mecanismo de iluminação infravermelha é preparado antes da aquisição da imagem. A câmera realiza a captura, o RTC fornece a data e a hora do evento e o arquivo JPEG é armazenado no cartão MicroSD. Em seguida, uma entrada é acrescentada ao arquivo CSV e o registro passa a fazer parte do histórico disponível na interface do equipamento.

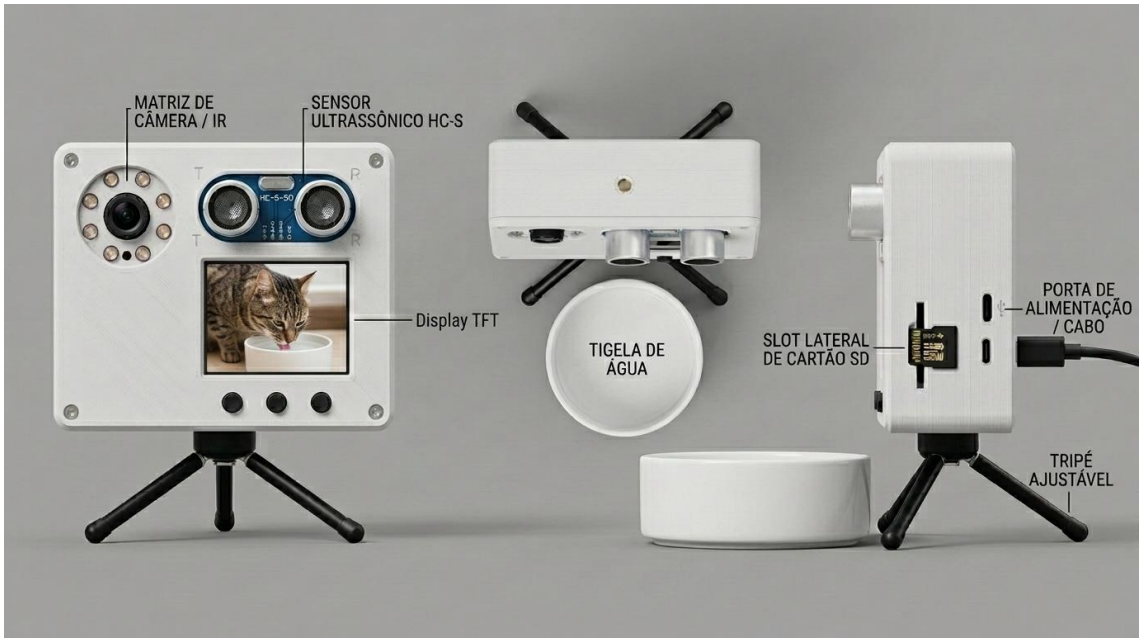
Essa organização estabelece uma cadeia de processamento que pode ser representada como:

detecção → confirmação → preparação da iluminação → captura → timestamp → armazenamento → registro CSV → atualização da IHM.

A arquitetura foi concebida para que cada etapa tenha uma responsabilidade definida. Essa separação facilita a manutenção do firmware e permite futuras ampliações, como sincronização via Wi-Fi, processamento de imagens e integração com aplicações externas.

A concepção física do HydraPet Vision 2.0 foi definida de modo que os elementos de sensoriamento, captura e interação com o usuário estejam reunidos em um único invólucro compacto, posicionado atrás do recipiente de água e voltado para a região de bebedouro. A Figura 25 apresenta as vistas frontal, superior e lateral do equipamento, com a identificação de seus principais elementos. Na face frontal estão dispostos, no canto superior esquerdo, a lente da câmera OV2640, circundada por um anel de LEDs infravermelhos, e, ao seu lado, o módulo ultrassônico HC-SR04, cujos transdutores de emissão (T) e de recepção (R) definem a zona de detecção; logo abaixo situam-se o display TFT e os três botões físicos de navegação. Na lateral, ficam o encaixe do cartão MicroSD e a porta USB tipo C de alimentação, enquanto um tripé ajustável permite regular a altura e a inclinação do conjunto em relação à tigela.

Figura 25 – Vistas frontal, superior e lateral do protótipo HydraPet Vision 2.0, com identificação dos principais elementos.



Fonte: Elaborado pelos autores (2026), com auxílio de ferramenta de inteligência artificial generativa.

A Figura 26 ilustra o equipamento em condição de uso no ambiente doméstico. O protótipo é posicionado ao lado e atrás do recipiente, de forma que o campo de visão da câmera e o feixe do sensor ultrassônico abranjam a região em que o gato se debruça para beber. Quando a presença do animal é confirmada pelo firmware, a fotografia capturada é gravada no cartão

MicroSD e exibida por alguns segundos no display, acompanhada da mensagem “FOTO SALVA” e do horário do registro.

Figura 26 – Concepção ilustrativa do protótipo HydraPet Vision 2.0 em operação junto ao recipiente de água.



Fonte: Elaborado pelos autores (2026), com auxílio de ferramenta de inteligência artificial generativa.

Ressalta-se que as Figuras 25 e 26 são ilustrações conceituais, geradas por ferramenta de inteligência artificial a partir da especificação do projeto. Elas não constituem fotografias de um equipamento fisicamente montado nem evidência de validação experimental, e detalhes como a quantidade de LEDs infravermelhos, as proporções do invólucro e o conteúdo exibido no display são meramente indicativos, prevalecendo as especificações descritas neste capítulo.

3.2 Detalhamento do Hardware e Conexões

A limitação de GPIOs da ESP32-CAM constitui uma das principais dificuldades de desenvolvimento do protótipo. A câmera OV2640 utiliza diversas linhas internas do ESP32, enquanto o leitor MicroSD também ocupa recursos de GPIO. Ao mesmo tempo, a versão 2.0 acrescentou três botões, sensor LDR, acionamento do ultrassom e controle da iluminação infravermelha.

Para solucionar essa limitação, foi utilizado o PCF8574, expansor de entradas e saídas digitais controlado por I2C. No firmware, o dispositivo é configurado no endereço 0x20, considerando A0, A1 e A2 conectados ao nível lógico baixo.

A utilização do PCF8574 permite concentrar diferentes sinais em um único dispositivo externo, reduzindo a quantidade de GPIOs diretamente ocupadas na ESP32-CAM.

3.2.1 Mapeamento de pinos da ESP32-CAM

A ESP32-CAM é uma placa de desenvolvimento compacta de baixo custo que atua como a unidade central de processamento e captura de imagem do protótipo. Ela funciona integrando um poderoso microcontrolador SoC ESP32-D0WDq6, que possui dois núcleos de 32 bits operando a até 240 MHz, com um conector dedicado para o sensor de imagem OV2640 de 2 Megapixel. O mapeamento utilizado pela versão 2.0 é apresentado na Tabela 2.

Figura 27 – ESP32-CAM



Fonte: Espressif

Tabela 2 – Mapeamento principal de pinos da ESP32-CAM no firmware 2.0

Componente	Função	GPIO / conexão	Observação
TFT ILI9341	SCK	GPIO 13	Clock SPI
TFT ILI9341	MOSI/SDI	GPIO 12	Dados SPI
TFT ILI9341	DC	GPIO 1	U0TXD reutilizado
TFT ILI9341	CS	GND	Único escravo considerado no barramento
TFT ILI9341	RST	3,3 V	Reset controlado por software
TFT ILI9341	LED	3,3 V	Backlight
MicroSD	SDMMC	GPIO 14, 15 e 2	Operação em 1 bit
HC-SR04	ECHO	GPIO 3	Entrada com interrupção
I2C	SDA	GPIO 26	Compartilhado com SCCB
I2C	SCL	GPIO 27	Compartilhado com SCCB
LED da placa	Flash	GPIO 4	Mantido em nível baixo
Câmera	XCLK	GPIO 0	Utilizado internamente pela câmera
PSRAM	—	GPIO 16	Não utilizado externamente

Fonte: Elaborado pelos autores (2026)

Além do pino XCLK (GPIO 0), a câmera OV2640 ocupa, por meio do barramento paralelo de dados e dos sinais de sincronismo, outras doze GPIOs da ESP32-CAM, conforme definido no firmware. A Tabela 3 detalha essa ocupação, evidenciando por que, somados aos pinos do display, do cartão MicroSD e do barramento I2C, restam poucas GPIOs livres na placa, o que fundamenta a adoção do expansor PCF8574 descrita na Seção 3.2.4.

Tabela 3 – Pinagem complementar do sensor de imagem OV2640 na ESP32-CAM

Sinal	GPIO(s)	Função
PWDN	GPIO 32	Power-down do sensor (uso interno)
RESET	—	Não conectado (RESET_GPIO_NUM = -1)
SIOD (SDA)	GPIO 26	SCCB compartilhado com o barramento I2C
SIOC (SCL)	GPIO 27	SCCB compartilhado com o barramento I2C
Y2 – Y9 (D0–D7)	GPIO 5, 18, 19, 21, 36, 39, 34, 35	Barramento paralelo de dados (8 bits)
VSYNC	GPIO 25	Sincronismo vertical de quadro
HREF	GPIO 23	Sincronismo horizontal de linha
PCLK	GPIO 22	Clock de pixel

Fonte: Elaborado pelos autores (2026)

A GPIO 1 merece atenção especial. Embora normalmente associada à transmissão serial U0TXD, na versão 2.0 ela é utilizada como sinal DC do display TFT. Por esse motivo, o firmware não inicializa a comunicação serial convencional para depuração. Essa decisão evita conflito entre a interface gráfica e a UART.

A GPIO 12 também exige atenção durante a montagem, pois participa da comunicação SPI com o display e é um pino associado a condições de boot do ESP32. Dessa forma, o circuito deve evitar a inclusão de resistores de pull-up inadequados nessa linha.

A GPIO 4 corresponde ao LED de flash integrado à própria placa ESP32-CAM. Como a versão 2.0 possui uma iluminação infravermelha externa controlada pelo PCF8574, esse LED não é utilizado pelo sistema. O firmware mantém a GPIO 4 em nível baixo para evitar acionamentos indesejados.

3.2.2 Display TFT ILI9341

A interface homem-máquina utiliza um display TFT de 2,4 polegadas equipado com controlador ILI9341. O display opera em resolução de 320 × 240 pixels no modo paisagem.

A comunicação ocorre por SPI através de uma instância HSPI configurada pelo firmware. Os sinais SCK e MOSI são conectados às GPIOs 13 e 12, respectivamente. O sinal DC utiliza a GPIO 1.

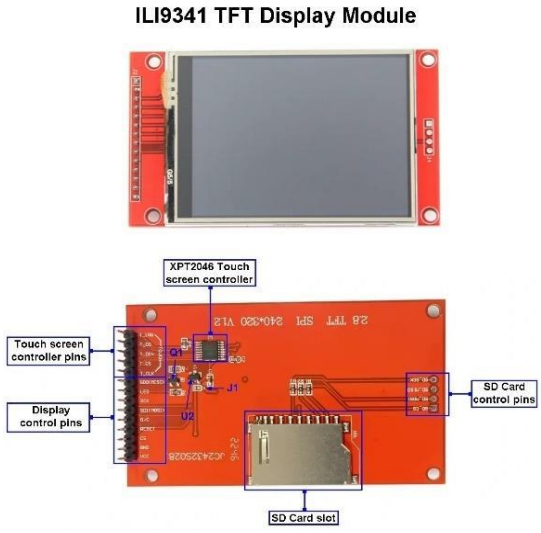
O terminal CS do display é conectado diretamente ao GND. Essa configuração é possível porque, na arquitetura atual, o display é tratado como o dispositivo ativo associado à interface SPI utilizada para a IHM. Dessa maneira, não é necessário reservar uma GPIO adicional exclusivamente para chip select.

O reset do display também não utiliza uma GPIO dedicada. O terminal RST permanece conectado a 3,3 V, sendo a inicialização realizada pelo próprio controlador por meio dos comandos enviados durante a configuração.

O firmware configura a velocidade do barramento SPI em 26 MHz e utiliza a biblioteca Adafruit ILI9341 em conjunto com a Adafruit GFX. A biblioteca TJpg_Decoder complementa esse conjunto ao permitir que arquivos JPEG sejam decodificados e enviados diretamente para a tela.

Essa integração é particularmente importante na função de visualização das fotografias. O sistema não precisa converter permanentemente os arquivos JPEG para outro formato antes de apresentá-los. O arquivo armazenado no cartão pode ser carregado e decodificado durante a consulta do histórico.

Figura 28 – Display TFT ILI9341



Fonte: Ilitek (ILI Technology Corp.)

3.2.3 Barramento I2C compartilhado

A comunicação I2C constitui um elemento fundamental da arquitetura da versão 2.0. As GPIOs 26 e 27 são utilizadas respectivamente como SDA e SCL. O barramento opera a 100 kHz, taxa do modo padrão da especificação I2C (NXP SEMICONDUCTORS, 2021), e é compartilhado entre o RTC e o PCF8574. Além desses dispositivos, as mesmas linhas estão associadas ao barramento SCCB utilizado pela câmera OV2640.

Essa configuração é possível porque o firmware foi desenvolvido especificamente para reutilizar o barramento. A opção de compilação SHARED_SCCB_BUS permanece habilitada e a configuração da câmera utiliza as linhas SCCB internas por meio do barramento já inicializado. O compartilhamento reduz a necessidade de novas GPIOs, aspecto particularmente importante em uma placa com disponibilidade limitada de pinos.

No barramento estão previstos os endereços:

- 0x68 para o RTC;
- 0x20 para o PCF8574.

Essa organização permite que diferentes periféricos sejam conectados às mesmas duas linhas físicas sem que cada componente necessite de um par exclusivo de GPIOs.

3.2.4 Expansor I/O PCF8574

O PCF8574 (NXP SEMICONDUCTORS, 2013) é utilizado como expansor de oito linhas digitais. No HydraPet Vision 2.0, seis das oito portas são utilizadas.

Tabela 4 – Distribuição das portas do PCF8574

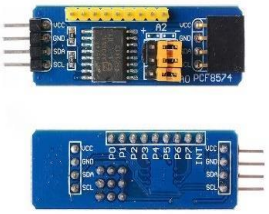
Porta	Função	Tipo de sinal
P0	Botão Avança	Entrada
P1	Botão Confirma	Entrada
P2	Botão Retrocede	Entrada
P3	TRIG do HC-SR04	Saída
P4	Gate do MOSFET dos LEDs IR	Saída
P5	DO do LDR/LM393	Entrada
P6	Reserva	Não utilizado
P7	Reserva	Não utilizado

Fonte: Elaborado pelos autores (2026).

As entradas dos botões e do LDR permanecem em nível lógico alto quando inativas e são interpretadas pelo firmware como sinais ativos em nível baixo. Essa característica é tratada diretamente pela função responsável pela leitura do PCF8574.

O estado das saídas é mantido pelo firmware por meio da variável pcfShadow. Isso é importante porque o PCF8574 trabalha com leitura e escrita das portas e, portanto, o software mantém uma representação do estado desejado para evitar alterações involuntárias nos demais bits durante a atualização de uma única saída.

Figura 29 – PCF8574: Módulo expansor de portas I2C



Fonte: Philips Semiconductors

3.2.5 Sensor ultrassônico HC-SR04

O HC-SR04 é empregado como sensor de proximidade; o módulo opera com 5 V e emite pulsos ultrassônicos de 40 kHz (ELECFREAKS, [s.d.]). Sua função no projeto não é medir diretamente o volume de água consumido, mas identificar a aproximação do animal à região de interesse. Na versão 2.0, o sinal TRIG do sensor é controlado pelo pino P3 do PCF8574. O sinal ECHO, por sua vez, é conectado à GPIO 3 da ESP32-CAM.

Como o HC-SR04 trabalha com sinal de retorno em nível lógico de aproximadamente 5 V, o ECHO deve passar por um divisor de tensão antes de alcançar a entrada do ESP32. No circuito adotado, utiliza-se um resistor de 1 kΩ em série e um resistor de 2 kΩ conectado ao GND. A medição é baseada no tempo de propagação do sinal ultrassônico. O firmware calcula a distância de acordo com:

$$d = (t \times 0{,}0343) / 2$$

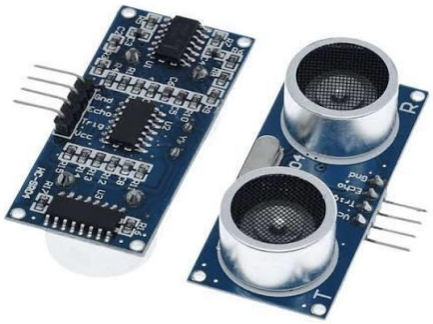
em que d representa a distância em centímetros e t representa o tempo de propagação do eco em microssegundos. O fator de divisão por dois ocorre porque o tempo medido corresponde ao percurso de ida e volta da onda. A principal evolução da versão 2.0 está no tratamento do ECHO. Em vez de utilizar a função pulseIn(), o firmware registra o início e o fim

do pulso por meio de uma interrupção configurada para detectar alterações no nível lógico da GPIO 3.

Quando o sinal passa para nível alto, a função de interrupção registra o instante por meio de micros(). Quando o sinal retorna para nível baixo, calcula-se a duração do pulso. O resultado é posteriormente convertido em distância no contexto normal de execução do programa. Essa abordagem reduz o bloqueio do processador associado a uma espera direta pelo retorno do sensor.

O firmware ainda estabelece um período máximo de espera de aproximadamente 35 ms. Caso o eco não seja recebido dentro desse intervalo, a leitura é considerada inválida e o sistema prossegue para as demais tarefas.

Figura 30 – Sensor ultrassônico HC-SR04



Fonte: Cytron Technologies

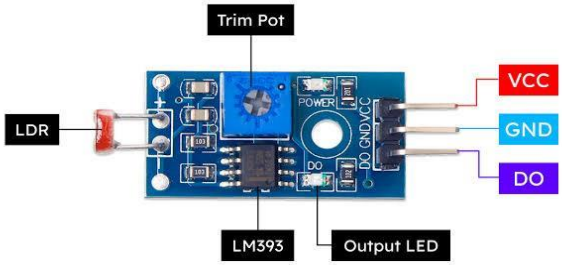
3.2.6 Sensor LDR e detecção de luminosidade

A versão 2.0 incorporou um sensor LDR associado a um comparador LM393 para identificar a condição luminosa do ambiente (TEXAS INSTRUMENTS, 2025). O sinal digital do módulo é conectado ao pino P5 do PCF8574. A leitura não é utilizada diretamente para acender ou apagar os LEDs a cada mudança instantânea. O firmware implementa uma janela de estabilidade de 3 segundos, definida pela constante LDR_STABLE_MS.

Esse mecanismo evita que pequenas oscilações de luminosidade produzam sucessivas mudanças no estado da iluminação. Quando uma mudança de estado é detectada, o firmware registra o novo estado como pendente e inicia a contagem de estabilidade. Somente quando a nova condição permanece durante o período estabelecido é que a variável interna isDark é efetivamente atualizada.

Esse procedimento representa uma forma simples de filtragem temporal do sinal do sensor, reduzindo oscilações provocadas por mudanças momentâneas de iluminação. A polaridade do LDR também pode ser configurada. O parâmetro ldrInvert permite inverter a interpretação do sinal digital, tornando o firmware adaptável a diferentes módulos cuja saída possa trabalhar com lógica invertida.

Figura 31 – Módulo Sensor de luz LDR, com saídas analógica e digital



Fonte: Joy-IT

3.2.7 Iluminação infravermelha de 850 nm

A iluminação noturna da versão 2.0 é composta por seis LEDs infravermelhos de aproximadamente 850 nm. Os LEDs são acionados por meio de um estágio de chaveamento com MOSFET. O PCF8574 fornece o comando para o gate do MOSFET através da saída P4. Um resistor de aproximadamente 100 Ω é utilizado na linha de acionamento do gate e um resistor de 100 kΩ mantém o gate referenciado ao GND na condição de desligamento.

Os seis LEDs são organizados em três ramos paralelos, cada um contendo dois LEDs em série e resistor limitador de corrente. Essa organização permite distribuir a corrente entre os ramos e proporcionar iluminação suficiente para a câmera durante condições de baixa luminosidade.

A arquitetura não utiliza o LED flash da própria ESP32-CAM. A iluminação infravermelha externa foi adotada justamente para possibilitar operação noturna sem depender de iluminação visível intensa.

O firmware oferece três modos de operação:

- automático, no qual a iluminação é determinada pela condição do ambiente;

• ligado, no qual o sistema mantém a iluminação infravermelha ativa;

• desligado, no qual os LEDs permanecem desativados.

No modo automático, quando a presença do animal se aproxima da zona de detecção em condição de baixa luminosidade, o firmware antecipa o acionamento da iluminação. Essa antecipação ocorre por meio da variável irHoldUntil e permite que os LEDs permaneçam ligados antes da captura.

Além disso, quando a fotografia é efetivamente solicitada e a iluminação ainda não está ativa, a função de captura pode ligar os LEDs e aguardar aproximadamente 300 ms antes da aquisição da imagem. Esse intervalo, definido por IR_PREHEAT_MS, tem a finalidade de permitir que a cena seja iluminada antes da captura.

Figura 32 – LEDs infravermelhos de 850 nm



Fonte: Epileds

3.2.8 Módulo RTC DS1307

O RTC DS1307 (ANALOG DEVICES, 2015) permanece como referência temporal do projeto. O módulo fornece ao firmware a data e a hora utilizadas para identificar cada evento. A comunicação ocorre pelo barramento I2C no endereço 0x68. A utilização de uma bateria de backup permite preservar a contagem de tempo durante interrupções da alimentação principal, desde que o módulo esteja adequadamente alimentado e a bateria esteja em condições de funcionamento.

Durante a inicialização, o firmware verifica a disponibilidade do RTC. Caso o dispositivo não seja encontrado, a variável rtcOk é marcada como falsa e a interface informa a condição ao usuário.

Quando o RTC está disponível, o firmware também verifica se o relógio está efetivamente em funcionamento. No caso do DS1307, a condição é avaliada pela função

isrunning(). Se o relógio tiver perdido a referência temporal, o firmware ajusta a data e a hora com base nos valores de compilação do programa e marca a condição como suspeita, permitindo que o usuário posteriormente realize o ajuste manual pela IHM. Essa funcionalidade representa uma evolução em relação à versão anterior, pois a versão 2.0 permite ao usuário acessar diretamente uma tela específica para configuração da data e da hora.

Figura 33 – Módulo RTC DS1307



Fonte: Analog Devices

3.2.9 Cartão MicroSD e armazenamento local

O cartão MicroSD constitui a memória não volátil utilizada para preservar as fotografias e os registros dos eventos. Na versão 2.0, o cartão é inicializado através da biblioteca SD_MMC em modo de 1 bit. Essa escolha é importante para reduzir a quantidade de GPIOs necessárias para a interface de armazenamento. O firmware cria automaticamente o diretório "/fotos" quando este ainda não existe. As fotografias são armazenadas nesse diretório.

O nome padrão de uma fotografia segue a estrutura:

- "/fotos/gato_AAAAMMDD_HHMMSS.jpg".

A utilização de ano, mês, dia, hora, minuto e segundo no nome do arquivo facilita a organização cronológica dos eventos. Existe ainda um tratamento para colisão de nomes. Caso já exista um arquivo com o mesmo nome, o firmware acrescenta um identificador baseado no valor de millis(), evitando a sobrescrita acidental de uma fotografia anterior.

Além das imagens, o sistema mantém o arquivo "/registros.csv". O cabeçalho utilizado pelo firmware é:

- data;hora;arquivo;distancia_cm;ir

Cada nova ocorrência registrada acrescenta uma linha contendo a data, a hora, o caminho da fotografia, a distância detectada e a indicação de utilização da iluminação

infravermelha. Essa estrutura amplia o valor do registro, pois o evento deixa de ser apenas uma fotografia isolada e passa a possuir informações de contexto.

Figura 34 – Cartão MicroSD



Fonte: SanDisk

3.3 Desenvolvimento do Firmware

O firmware da versão 2.0 do protótipo HydraPet Vision foi construído na linguagem C++, utilizando o ecossistema e a estrutura de compilação do ambiente Arduino IDE voltado à arquitetura de microcontroladores ESP32. A concepção do software priorizou o determinismo na execução das rotinas de monitoramento comportamental, a redução de operações bloqueantes, com a consequente melhoria da responsividade global do sistema, e a estruturação modular das tarefas de Interface Homem-Máquina (IHM), sensoriamento, iluminação auxiliar infravermelha, captura de imagem e persistência de dados. O controle de versionamento do código é gerenciado internamente por meio da constante FW_VERSION, fixada no valor "2.0.0".

A modularização e a abstração do hardware foram viabilizadas pelo emprego de bibliotecas especializadas. A seguir, apresentam-se as fundamentações técnicas e as justificativas funcionais para a inclusão de cada uma das bibliotecas utilizadas no projeto.

3.3.1 Biblioteca Arduino.h

A biblioteca Arduino.h constitui a base da aplicação desenvolvida para o HydraPet Vision 2.0, disponibilizando as principais funções utilizadas na programação do microcontrolador ESP32-CAM dentro do ambiente Arduino. Entre seus recursos estão as funções de controle das entradas e saídas digitais, como pinMode(), digitalRead() e

digitalWrite(), além das funções relacionadas à temporização, como millis(), micros(), delay() e delayMicroseconds().

No firmware desenvolvido para a versão 2.0, a Arduino.h está diretamente relacionada à estrutura convencional de execução formada pelas funções setup() e loop(). A função setup() é responsável pela inicialização dos periféricos, sensores, câmera, cartão MicroSD, RTC, display e demais recursos necessários ao funcionamento do sistema. Após essa etapa, a função loop() passa a executar continuamente as rotinas responsáveis pelo monitoramento e pelo controle do protótipo.

A organização do firmware foi desenvolvida de forma cooperativa, utilizando rotinas periódicas para distribuir as diferentes funções do sistema sem a necessidade de criar tarefas independentes por meio do FreeRTOS. Funções como sensorTask(), irTask() e uiTask() são chamadas durante a execução do loop() e realizam suas respectivas operações de acordo com os intervalos de tempo definidos no programa. Essa abordagem permite organizar o processamento dos sensores, o controle da iluminação infravermelha e a atualização da interface mantendo uma estrutura de firmware simples e adequada aos recursos disponíveis na ESP32-CAM.

Para o controle temporal dessas rotinas, são utilizadas principalmente funções como millis(), permitindo verificar a passagem de intervalos de tempo sem interromper continuamente a execução do programa. Essa estratégia é aplicada, por exemplo, no controle do período de leitura do sensor ultrassônico, na confirmação das leituras de presença, no acionamento da iluminação infravermelha, no tratamento dos botões e no gerenciamento de temporizações da interface.

A medição do sensor HC-SR04 utiliza ainda uma abordagem orientada por interrupção de hardware para o sinal de ECHO, conectado ao GPIO 3. Nesse caso, a função attachInterrupt() permite identificar as mudanças de estado do sinal e registrar os instantes correspondentes ao início e ao término do pulso. Dessa forma, a medição da distância pode ocorrer sem a utilização de uma rotina bloqueante para aguardar o retorno do sinal ultrassônico.

Assim, na versão 2.0 do HydraPet Vision, a biblioteca Arduino.h fornece a infraestrutura básica para o controle do microcontrolador e para a organização temporal do firmware. A combinação entre o ciclo setup()/loop(), temporizações baseadas em millis() e interrupções de hardware permite implementar o comportamento necessário ao protótipo sem utilizar tarefas FreeRTOS criadas explicitamente pelo programa. Essa arquitetura mantém o firmware organizado e possibilita a execução integrada das funções de sensoriamento, captura, armazenamento e interface do sistema.

3.3.2 Biblioteca Wire.h

A biblioteca Wire.h é o componente padrão para a implementação e o gerenciamento da comunicação serial síncrona baseada no protocolo I2C (Inter-Integrated Circuit). Ela abstrai os registradores físicos dos barramentos do ESP32, permitindo a configuração das linhas de dados (Serial Data – SDA) e de relógio (Serial Clock – SCL), o ajuste da taxa de transmissão (definida em 100 kHz) e a execução de transações do tipo mestre-escravo. No protótipo HydraPet Vision v2.0, a Wire.h desempenha um papel central na arquitetura de expansão de hardware. Como o SoC ESP32-CAM possui um número extremamente reduzido de pinos GPIO disponíveis, a comunicação I2C foi adotada para centralizar múltiplos periféricos no mesmo barramento serial, especificamente os pinos GPIO 26 (SDA) e GPIO 27 (SCL). Por meio da Wire.h, o firmware realiza a leitura e a escrita no expansor de I/O de 8 bits PCF8574, no relógio de tempo real (RTC) e na interface de controle da câmera (SCCB), otimizando a transmissão de bytes e eliminando o consumo desnecessário de pinos do microcontrolador.

3.3.3 Biblioteca SPI.h

A biblioteca SPI.h viabiliza o controle e a troca de dados através do protocolo serial síncrono Serial Peripheral Interface (SPI), caracterizado pelo funcionamento em modo full-duplex e por elevadas taxas de transferência de informações. A comunicação fundamenta-se nas linhas de relógio (SCK), entrada de dados (MISO), saída de dados (MOSI) e seleção de escravo (CS). No firmware do HydraPet Vision v2.0, a SPI.h é empregada exclusivamente para a navegação e atualização gráfica do display colorido TFT ILI9341 de 2,4 polegadas. Considerando que a renderização de telas, menus e imagens exige o envio massivo de pixels codificados em RGB565 (16 bits por pixel), a utilização da interface SPI configurada em 26 MHz frequência definida pela chamada tft.begin(26000000) na inicialização do sistema, é indispensável para garantir taxas de atualização fluidas e sem artefatos visuais, dentro do limite confiável de operação do controlador ILI9341 nesse barramento. A biblioteca gerencia o temporizador do barramento e os pacotes de dados enviados ao controlador gráfico, estabelecendo um canal dedicado de comunicação de alta velocidade a partir das portas GPIO 12 e GPIO 13 do ESP32, configuradas no firmware a 26 MHz.

3.3.4 Biblioteca Adafruit_GFX.h

A biblioteca Adafruit_GFX.h fornece um conjunto abrangente de primitivas gráficas bidimensionais e rotinas de renderização de texto, atuando como um motor gráfico universal independente do tipo de tela utilizado. Ela define algoritmos otimizados para o desenho de pontos, linhas, retângulos, círculos, preenchimentos de blocos de cor e gerenciamento de fontes tipográficas com diferentes escalas e alinhamentos. No HydraPet Vision v2.0, a Adafruit_GFX.h serve como camada base para a construção da Interface Homem-Máquina (IHM). É por meio de suas funções que o firmware desenha os elementos estáticos e dinâmicos do painel do usuário, tais como a barra de cabeçalho, os quadros divisores de status, as caixas de seleção destacadas do menu interativo, as mensagens de alerta e os indicadores de estado (SD, IR e luminosidade). A abstração fornecida pela biblioteca permite manipular coordenadas cartesianas (X, Y) na resolução de 320 × 240 pixels com alto desempenho de processamento, simplificando a programação de telas complexas.

3.3.5 Biblioteca Adafruit_ILI9341.h

A biblioteca Adafruit_ILI9341.h é o driver específico para a comunicação e o controle do circuito integrado ILI9341 (ILI TECHNOLOGY, [s.d.]), responsável pela varredura e exibição de imagens em matrizes de cristal líquido TFT (320x240 pixels RGB). Herdeira das funções genéricas da Adafruit_GFX.h, esta biblioteca traduz os comandos de desenho geométrico em sequências de registradores e envios de sinal aceitos pelo hardware do ILI9341 através da interface SPI. No projeto, ela responde pelo envio dos comandos de inicialização da tela, pela definição da orientação espacial de exibição (setRotation), pela configuração do formato cromático e pelo preenchimento da tela, sem alocar framebuffer na RAM do ESP32, pois o ILI9341 mantém a memória de imagem no próprio controlador (fillScreen). Além disso, em conjunto com a porta lógica atribuída ao sinal de Controle/Dados (DC), a biblioteca garante que o microcontrolador alterne corretamente entre o envio de instruções de configuração e blocos diretos de pixels de dados de cor, permitindo a exibição das métricas do sistema e dos arquivos da galeria.

3.3.6 Biblioteca RTClib.h

A biblioteca RTClib.h (desenvolvida pela Adafruit) fornece abstrações orientadas a objetos para a manipulação de relógios de tempo real (RTC – Real-Time Clock), com suporte

nativo aos circuitos integrados DS1307 e DS3231 via barramento I2C. Ela gerencia o formato do tempo, a conversão de registros precisos de data e hora (timestamps) e o cálculo do calendário gregoriano, incluindo a contagem automática de anos bissextos. No HydraPet Vision v2.0, a precisão da marcação de tempo é um requisito fundamental para a validade dos dados comportamentais do felino. A RTClib.h é utilizada tanto no momento da inicialização do sistema para verificar o funcionamento do cristal oscilador do RTC quanto durante a execução da rotina de captura fotográfica. Quando o sensor detecta a aproximação do animal, a biblioteca fornece a data e a hora exatas (ano, mês, dia, hora, minuto e segundo), as quais são convertidas em texto padronizado para a composição do nome único do arquivo de imagem e para o registro de data/hora gravado no log histórico em formato CSV.

3.3.7 Biblioteca TJpg_Decoder.h

A biblioteca TJpg_Decoder.h (desenvolvida por Bodmer; BODMER, [s.d.]) é um motor de descompressão de dados fotográficos em formato JPEG (padrão ITU-T T.81; INTERNATIONAL TELECOMMUNICATION UNION, 1992) otimizado para microcontroladores de recursos limitados. Ela implementa o algoritmo Tiny JPEG Decompressor, capaz de ler fluxos de bytes comprimidos a partir da memória Flash ou do cartão de memória e reconstruir a matriz de imagem em blocos mínimos codificados (MCU – Minimum Coded Unit), convertendo-os diretamente para o formato de cor RGB565 aceito pelos displays TFT. No protótipo HydraPet Vision v2.0, a TJpg_Decoder.h viabiliza a criação de uma galeria de fotos local diretamente na tela do dispositivo. Em vez de exigir o envio das fotos para um computador ou smartphone, o tutor ou o médico-veterinário pode navegar pelos arquivos do cartão MicroSD e visualizar as imagens capturadas na tela de 2,4 polegadas. A biblioteca gerencia a alocação do buffer de descompressão, a inversão da ordem dos bytes de cor (byte-swapping) e a chamada de callbacks de renderização por blocos, otimizando o uso da memória RAM.

3.3.8 Biblioteca Preferences.h

A biblioteca Preferences.h é uma camada de abstração fornecida pelo ambiente do ESP32 para o gerenciamento de dados na memória Flash não volátil (NVS – Non-Volatile Storage). Ela utiliza um sistema de armazenamento estruturado em chave-valor sob a forma de tabela Hash, substituindo o antigo conceito de EEPROM por uma solução com suporte a múltiplos espaços de nomes (namespaces), resistência a falhas de energia e mecanismo de

nivelamento de desgaste (wear leveling) do chip Flash. No HydraPet Vision v2.0, a Preferences.h é responsável pela persistência das configurações do sistema ajustadas pelo usuário na IHM. Parâmetros como a distância limiar do ultrassom (chave thr), o intervalo entre capturas (cool), o modo de operação da iluminação infravermelha (ir), a polaridade do sinal digital do módulo LDR (ldr) e o número de leituras de confirmação (conf) são gravados na memória NVS. Isso garante que, mesmo em situações de corte súbito de energia, as preferências do usuário permaneçam salvas e sejam restauradas automaticamente ao ligar o protótipo.

3.3.9 Biblioteca driver/gpio.h

A biblioteca driver/gpio.h faz parte do kit de desenvolvimento de software nativo da fabricante do microcontrolador (ESP-IDF - Espressif IoT Development Framework) e fornece acesso direto às funções de controle de baixo nível das portas de entrada e saída de propósito geral (GPIO). Ela permite configurar modos de operação avançados, resistores internos de pull-up e pull-down, matrizes de sinal, níveis de interrupção por hardware e atribuição de rotinas de serviço de interrupção (ISR – Interrupt Service Routine) alocadas na memória RAM de alta velocidade (IRAM_ATTR). No HydraPet Vision v2.0, o uso da driver/gpio.h limita-se à leitura rápida do nível do pino (gpio_get_level) dentro da rotina de interrupção; a conversão da leitura do sensor ultrassônico HC-SR04 de um modelo bloqueante para um modelo por interrupção decorre de attachInterrupt() e de echoISR(). O firmware associa uma interrupção por mudança de estado ao pino de retorno ECHO (GPIO 3), permitindo registrar com resolução de microssegundos o tempo do pulso sem paralisar o processamento do núcleo do ESP32, garantindo a execução contínua de outras tarefas do sistema.

3.3.10 Biblioteca esp_camera.h

A biblioteca esp_camera.h é o driver oficial da Espressif para o gerenciamento de sensores de imagem (como o OV2640) integrados aos módulos da família ESP32. Ela configura o barramento paralelo DVP de 8 bits, os sinais de sincronismo vertical e horizontal (VSYNC, HREF), o relógio do sistema de visão (XCLK), o canal DMA (Direct Memory Access) e o controlador do receptor de câmera do microcontrolador. No firmware do HydraPet Vision v2.0, a biblioteca esp_camera.h abstrai toda a complexidade do hardware de captura de vídeo. Ela inicializa o sensor OV2640 (OMNIVISION, 2006) com a resolução QVGA (320 × 240 pixels), ajusta os níveis de qualidade da compressão JPEG, aloca buffers de captura na memória PSRAM externa do ESP32-CAM e gerencia a aquisição do ponteiro da estrutura camera_fb_t.

Essa estrutura contém o buffer de bytes da fotografia pronta para ser gravada no cartão MicroSD ou processada pelo sistema, com os buffers alocados em PSRAM, o que preserva a memória interna.

3.3.11 Bibliotecas FS.h e SD_MMC.h

As bibliotecas FS.h e SD_MMC.h trabalham em conjunto para fornecer a infraestrutura do sistema de arquivos e a interface de comunicação física com cartões de memória MicroSD formatados no padrão FAT/FAT32. A FS.h define a classe genérica de manipulação de diretórios e arquivos (File), fornecendo métodos padronizados para leitura, escrita, abertura, fechamento e busca de ponteiros (open, write, read, printf). Por sua vez, a SD_MMC.h implementa o driver de hardware para o controlador SDMMC nativo do ESP32, operando em modo de barramento de 1 bit para economizar pinos de entrada e saída. No HydraPet Vision v2.0, essas bibliotecas gerenciam o armazenamento persistente de dados no cartão de memória. Elas são responsáveis por criar a estrutura de diretórios (/fotos), gravar os arquivos de imagem compactados em formato JPEG obtidos pela câmera e manter atualizado o banco de dados local /registros.csv, que armazena o histórico textual de todos os acessos do felino para posterior análise.

3.3.12 Inicialização do sistema

A inicialização começa com a configuração da GPIO 4 como saída e sua manutenção em nível baixo. Essa etapa garante que o flash integrado da ESP32-CAM permaneça desligado. Em seguida, o firmware configura a interface SPI HSPI utilizada pelo display. O clock é estabelecido na GPIO 13 e os dados MOSI na GPIO 12. O display é iniciado a 26 MHz e configurado para orientação paisagem. Depois da inicialização gráfica, o firmware carrega as configurações persistentes através da função cfgLoad().

Caso não existam configurações armazenadas, são carregados os valores padrão:

- distância de detecção: 15 cm;
- intervalo entre fotografias: 10 s;
- iluminação IR: automática;
- polaridade do LDR: normal;
- confirmação de presença: quatro leituras consecutivas

Na sequência, o barramento I2C é iniciado nas GPIOs 26 e 27 com frequência de 100 kHz. O firmware inicializa a câmera utilizando a configuração específica da AI-Thinker ESP32-CAM. O clock XCLK é configurado em 20 MHz e o formato da imagem é JPEG.

Quando a PSRAM está disponível, o sistema utiliza dois framebuffers localizados na PSRAM, permitindo uma operação mais adequada para aquisição de imagens. A resolução é configurada como QVGA, ou seja, 320 × 240 pixels, com qualidade JPEG configurada para 10.

A câmera também recebe configurações relacionadas à exposição automática, ganho, balanço de branco e correção de lente. O controle AEC2 é habilitado para favorecer a captura em cenas com pouca iluminação, característica particularmente relevante para a operação com iluminação infravermelha.

Depois da câmera, o firmware verifica a presença do PCF8574. Caso o expansor responda no endereço configurado, o sistema escreve o estado inicial das portas, mantendo a iluminação infravermelha desligada e o TRIG do sensor em estado baixo.

O RTC é então inicializado. A disponibilidade do relógio é registrada na variável rtcOk. Em seguida, o cartão MicroSD é montado. Caso a inicialização seja bem-sucedida, o sistema verifica a existência do diretório de fotografias e carrega os registros existentes no arquivo CSV.

Por fim, o pino ECHO do HC-SR04 é configurado como entrada e a interrupção é associada à GPIO 3. Depois dessas etapas, o sistema realiza uma leitura inicial do LDR, aguarda a estabilização da inicialização e direciona a interface para a tela principal.

3.3.13 Configuração persistente por NVS

Uma das principais evoluções do firmware 2.0 é a utilização da memória não volátil NVS, implementada pela biblioteca Preferences. As configurações são armazenadas no namespace “hydrapet”. Os seguintes parâmetros são persistidos:

- “thr”: distância de detecção;
- “cool”: intervalo entre fotografias;
- “ir”: modo de iluminação;
- “ldr”: polaridade do LDR;
- “conf”: quantidade de leituras consecutivas para confirmação.

A função cfgLoad() carrega os parâmetros durante a inicialização. A função cfgSave() grava as alterações realizadas pelo usuário.

Essa arquitetura evita que o operador precise reconfigurar o sistema toda vez que a alimentação for interrompida.

O recurso também torna o protótipo mais adequado para testes experimentais, pois diferentes condições de operação podem ser configuradas diretamente pela IHM e mantidas entre reinicializações.

3.4 Sensoriamento de Presença e Confirmação do Evento

O sensoriamento de presença constitui uma das etapas fundamentais do funcionamento do HydraPet Vision, pois sua finalidade não é apenas identificar a presença de um objeto próximo ao dispositivo, mas determinar se essa aproximação apresenta consistência suficiente para ser considerada um evento válido e, consequentemente, iniciar uma captura fotográfica. No firmware versão 2.0, essa função é realizada pela rotina sensorTask(), que trabalha com período nominal de 60 ms, definido pela constante SENSOR_PERIOD_MS.

Quando não há uma medição em andamento, o firmware verifica se esse intervalo já foi transcorrido desde o último acionamento do sensor. Em caso afirmativo, o estado do sinal ECHO é reiniciado e o sinal TRIG é acionado por meio do expansor de entradas e saídas PCF8574. Após esse acionamento, o sistema passa para o estado waitingEcho, permanecendo disponível para o processamento das demais funções do firmware enquanto aguarda o retorno do sensor.

Dessa forma, a medição não depende de uma espera bloqueante, uma vez que o monitoramento do sinal ECHO é realizado pela rotina de interrupção echoISR(), associada à GPIO 3 da ESP32-CAM. Durante a medição, essa rotina identifica as transições do sinal ECHO. Na borda de subida, o instante é armazenado na variável echoRise; posteriormente, na borda de descida, a diferença entre o instante atual e o valor armazenado é calculada e registrada em echoWidth.

A rotina sensorTask() verifica então a disponibilidade do resultado por meio da variável echoDone e, quando a medição está concluída, converte a duração do pulso em distância. Caso não seja identificado um retorno dentro de aproximadamente 35 ms, a medição é encerrada como inválida. A distância obtida é encaminhada à função processDistance(), responsável por avaliar as condições de presença.

Inicialmente, o valor é comparado com a faixa mínima válida, estabelecida em 2 cm pela constante MIN_VALID_CM, e com o limite de detecção configurado pelo usuário. Paralelamente, o firmware considera uma faixa ampliada, correspondente ao dobro da distância configurada, utilizada como condição antecipada para a preparação da iluminação infravermelha.

Quando o objeto se encontra nessa região ampliada e o sistema identifica baixa luminosidade, é mantida uma janela de preparação da iluminação infravermelha, permitindo que o IR esteja disponível quando a captura for efetivamente solicitada. Esse mecanismo contribui para que a iluminação seja preparada antes da aquisição da imagem, especialmente em situações nas quais o ambiente apresenta pouca luminosidade.

Para evitar que uma única leitura seja suficiente para disparar uma fotografia, o firmware utiliza um mecanismo de confirmação baseado em leituras consecutivas. Cada medição válida dentro da faixa configurada incrementa a variável hits; quando uma leitura deixa essa faixa, o contador é reiniciado.

A captura somente é solicitada quando o número de leituras consecutivas atinge ou ultrapassa o valor definido em cfg.confirmReads. Na configuração padrão do sistema, são necessárias quatro leituras consecutivas válidas para confirmar a presença. Esse procedimento aumenta a estabilidade do sensoriamento, pois uma leitura isolada pode ocorrer em razão de reflexões, movimentações rápidas ou da presença momentânea de outros objetos na região monitorada.

Após a confirmação da presença, o firmware ainda verifica o intervalo mínimo estabelecido entre duas fotografias. A variável lastPhoto registra o instante da última captura, enquanto cfg.cooldownS determina o período de espera configurado, cujo valor padrão é de 10 segundos.

Assim, mesmo que o animal permaneça próximo ao recipiente, o sistema não realiza uma sequência contínua de fotografias. Esse mecanismo reduz a redundância dos registros e evita que uma única permanência do felino junto ao bebedouro seja interpretada como vários eventos independentes, contribuindo para a organização do histórico fotográfico e para o funcionamento mais consistente do protótipo.

3.5 Controle Automático da Iluminação Infravermelha

A iluminação infravermelha constitui um dos recursos responsáveis pela operação do HydraPet Vision em condições de baixa luminosidade, permitindo que o sistema mantenha a

capacidade de registro fotográfico mesmo durante períodos noturnos ou em ambientes com pouca iluminação. No firmware versão 2.0, o controle da iluminação considera três informações principais: o modo de operação selecionado pelo usuário, a condição luminosa identificada pelo sensor LDR e o estado atual dos LEDs infravermelhos. Essa organização permite que o acionamento seja realizado de maneira automática ou manual, conforme a configuração estabelecida na interface do equipamento.

No modo automático (IR_AUTO), a condição luminosa do ambiente é determinada pelo módulo LDR associado ao comparador LM393. Para evitar que variações momentâneas da iluminação provoquem acionamentos sucessivos, o firmware utiliza um período de estabilidade de aproximadamente três segundos antes de atualizar a variável isDark. Dessa forma, a condição de baixa luminosidade somente é considerada válida quando permanece estável pelo intervalo definido em LDR_STABLE_MS, contribuindo para um comportamento mais consistente do sistema.

A rotina irTask() verifica continuamente qual deve ser o estado da iluminação. Quando o usuário seleciona IR_ON, os LEDs permanecem acionados; quando seleciona IR_OFF, permanecem desligados. No modo automático, por sua vez, o acionamento depende da condição luminosa e das condições de preparação estabelecidas pelo sistema. A alteração efetiva do estado é realizada pela função setIR(), que somente envia uma nova atualização ao PCF8574 quando o estado solicitado é diferente daquele que já está aplicado.

Além do acionamento baseado diretamente na luminosidade, o firmware possui uma estratégia de preparação antecipada da iluminação. Quando o animal entra em uma região correspondente a até duas vezes o limite de detecção configurado e o ambiente está escuro, o sistema estabelece uma janela de preparação de aproximadamente 2,5 segundos. Essa condição permite que a iluminação infravermelha seja preparada antes que a presença seja confirmada definitivamente pelo mecanismo de leituras consecutivas.

Quando a captura é efetivamente solicitada, a função doCapture() realiza uma nova verificação da necessidade de iluminação. Caso o IR seja necessário e os LEDs ainda estejam desligados, o firmware os aciona e aguarda aproximadamente 300 ms, período definido por IR_PREHEAT_MS, antes de realizar a aquisição da imagem. Essa etapa evita que a fotografia seja registrada imediatamente após o acionamento dos LEDs, proporcionando um intervalo para que a iluminação esteja disponível no momento da captura.

A câmera também possui recursos próprios de controle de exposição e ganho, utilizados em conjunto com a iluminação infravermelha. Entre as configurações aplicadas ao sensor OV2640 estão o controle automático de exposição e o recurso AEC2, além do controle

automático de ganho e do limite de ganho configurado no firmware. A combinação entre essas funções da câmera e o acionamento controlado do IR permite adequar o processo de aquisição às condições de luminosidade previstas para o funcionamento do protótipo.

3.6 Processo de Captura Fotográfica

A captura fotográfica é iniciada somente quando a variável captureRequested assume valor verdadeiro, condição estabelecida após a confirmação da presença pelo sistema de sensoriamento e a verificação do intervalo mínimo entre eventos. Quando a solicitação é identificada, a função doCapture() inicia o procedimento, limpa a solicitação pendente e registra o instante associado ao processo de captura. Essa organização permite separar a etapa de detecção da etapa de aquisição da imagem, mantendo uma sequência lógica entre os diferentes subsistemas do protótipo.

Antes de acessar a câmera, o firmware verifica as condições camOk e sdOk, que representam a disponibilidade da câmera e do cartão MicroSD, respectivamente. Caso seja identificada alguma falha, a interface do equipamento informa a condição ao usuário, evitando que o sistema prossiga com uma operação que não poderá ser concluída corretamente. Quando a condição de iluminação exige o uso do infravermelho, o firmware também garante que os LEDs estejam acionados antes de iniciar a aquisição da imagem.

Na sequência, a câmera realiza uma primeira aquisição de quadro que é descartada pelo firmware. Esse procedimento tem a finalidade de evitar que um quadro anteriormente disponível no buffer seja utilizado como se correspondesse ao novo evento detectado. Após essa etapa, um novo quadro é solicitado ao sensor OV2640. A imagem é capturada em formato JPEG e resolução QVGA, correspondente a 320 × 240 pixels, conforme definido na configuração da câmera utilizada pelo projeto.

O quadro capturado é disponibilizado por meio de um framebuffer gerenciado pela biblioteca da câmera. Quando a PSRAM está disponível na ESP32-CAM, os buffers de imagem são alocados nessa memória, permitindo o gerenciamento dos dados fotográficos com maior espaço de armazenamento temporário. Após a aquisição do quadro válido, o firmware consulta o relógio de tempo real por meio da função getNow(true), utilizando o parâmetro true para solicitar uma leitura atualizada do RTC e obter a referência temporal correspondente ao evento.

A partir dessas informações, são definidos a data, o horário e o caminho de armazenamento da fotografia. A data é organizada no formato AAAA-MM-DD, enquanto o horário utiliza o padrão HH:MM:SS. O nome do arquivo segue a estrutura

gato_AAAAMMDD_HHMMSS.jpg, sendo armazenado no subdiretório /fotos/AAAA-MM-DD do cartão MicroSD. A utilização da data e do horário no nome do arquivo facilita a organização cronológica das fotografias e permite associar visualmente o arquivo ao momento em que o evento foi registrado.

Antes da gravação, o firmware verifica se já existe um arquivo com o mesmo nome. Caso seja encontrada uma ocorrência, é acrescentado um identificador adicional baseado no tempo de execução do sistema, evitando a substituição acidental de uma fotografia existente. Em seguida, o arquivo é aberto no cartão MicroSD em modo de escrita e o conteúdo binário armazenado no framebuffer é gravado diretamente no arquivo correspondente.

Após a conclusão da gravação da imagem, o firmware atualiza o banco de registros por meio da função appendCsv(). Uma nova linha é acrescentada ao arquivo /registros.csv, contendo a data, a hora, o caminho da fotografia, a distância identificada no momento do evento e a informação referente ao uso da iluminação infravermelha. Dessa maneira, cada fotografia permanece associada aos principais parâmetros registrados pelo sistema durante sua aquisição.

O registro também é incorporado à estrutura mantida em memória RAM por meio da função addLogEntry(). Nesse momento, o contador totalEvents é atualizado e a quantidade de registros realizados no dia é recalculada. Ao final do procedimento, o framebuffer utilizado pela câmera é devolvido por meio da função esp_camera_fb_return(), liberando o recurso de memória para que novas aquisições possam ser realizadas.

Quando a captura ocorre a partir da tela principal, o sistema apresenta uma pré-visualização da fotografia durante aproximadamente três segundos. A interface informa que a foto foi salva e apresenta o horário correspondente ao registro, indicando também quando a captura foi realizada com utilização do modo infravermelho. Esse recurso fornece ao usuário uma confirmação visual imediata de que o evento foi processado e armazenado pelo equipamento.

3.7 Banco de Registros e Organização dos Dados

A versão 2.0 apresenta uma estrutura de registros mais organizada, permitindo relacionar cada fotografia aos parâmetros associados ao evento de detecção. O arquivo CSV utilizado pelo sistema funciona como um banco de eventos local, adotando uma estrutura simples e legível, que pode ser posteriormente processada por softwares externos para análise dos dados. Essa abordagem mantém os registros em formato aberto e evita a necessidade de um sistema de banco de dados complexo no próprio dispositivo.

O arquivo /registros.csv utiliza, na revisão 2.1, o cabeçalho data;hora;arquivo;distancia_cm;ir;visita (a versão 2.0 não tinha a última coluna). Dessa forma, cada linha reúne as informações necessárias para relacionar temporalmente o evento à fotografia correspondente. Um registro pode ser representado, por exemplo, como 2026-09-20;14:32:18;/fotos/2026-09-20/gato_20260920_143218.jpg;13,4;1;57, no qual são armazenados a data e o horário do evento, o caminho da imagem, a distância medida no momento da detecção e o estado da iluminação infravermelha.

Durante a inicialização do sistema, o conteúdo do arquivo CSV é carregado por meio da função loadLog(). O firmware percorre as linhas armazenadas no cartão MicroSD, desconsidera o cabeçalho e separa os campos utilizando o caractere ; como delimitador. As informações obtidas são então organizadas em estruturas do tipo LogEntry, permitindo que os registros sejam utilizados posteriormente pela interface de consulta e visualização do histórico fotográfico.

Para possibilitar uma navegação rápida pelos registros, o firmware mantém em memória RAM até 150 eventos recentes, conforme o limite definido pela constante MAX_LOG. Quando essa capacidade é atingida, os registros mais antigos mantidos na RAM são deslocados e o novo evento é inserido na estrutura. Esse procedimento está relacionado apenas à memória de trabalho utilizada pela interface e não representa a exclusão dos arquivos ou registros armazenados de forma permanente no cartão MicroSD.

O armazenamento persistente permanece, portanto, no cartão MicroSD, onde as fotografias são mantidas em subpastas por data dentro de /fotos e o histórico de fotos no arquivo /registros.csv. Essa separação entre armazenamento permanente e memória de trabalho constitui uma característica importante da arquitetura do sistema, pois permite preservar os dados mesmo após o desligamento do equipamento, enquanto a RAM é utilizada para proporcionar acesso mais rápido aos registros durante a operação da interface.

A organização adotada também favorece a rastreabilidade dos eventos, uma vez que cada entrada do arquivo CSV contém uma referência direta à fotografia correspondente. Com isso, o sistema consegue relacionar o momento da detecção, a distância registrada e a condição de iluminação à imagem armazenada, formando um histórico estruturado dos eventos monitorados pelo HydraPet Vision.

3.8 Interface Homem-Máquina

A Interface Homem-Máquina (IHM) do HydraPet Vision 2.0 foi desenvolvida para possibilitar a interação do usuário com as funcionalidades de monitoramento e registro fotográfico de eventos associados à aproximação de um animal ao recipiente de água. A interface foi implementada em firmware para um display TFT ILI9341 de 2,4 polegadas, com resolução de 320 × 240 pixels, e utiliza três botões físicos conectados ao expansor de entradas e saídas PCF8574.

A estrutura da interface é organizada em sete estados de tela: principal, menu, registros de fotos, visualização de fotografia, ajuste de data e hora, configurações e informações do sistema. Essa organização permite separar as funções de acompanhamento do sistema, consulta das fotografias registradas e parametrização do funcionamento, proporcionando uma navegação estruturada entre as funcionalidades implementadas.

A tela principal apresenta informações sobre o estado operacional do sistema, incluindo a condição do cartão microSD, o modo de iluminação infravermelha, a condição de luminosidade identificada pelo módulo LDR e indicadores relacionados à quantidade de registros e à distância detectada. Também são exibidas mensagens de acompanhamento, como “Monitorando...” e “Gato detectado!”, conforme as condições lógicas identificadas pelo firmware. A apresentação desses indicadores depende da disponibilidade dos componentes e das informações obtidas durante a execução do programa.

O menu principal disponibiliza cinco opções: “Registros de fotos”, “Ajustar data e hora”, “Configuracoes”, “Info do sistema” e “Voltar ao inicio”. A consulta das fotografias é realizada pela opção de registros. Nessa tela, o firmware apresenta as entradas disponíveis para seleção e permite acessar a visualização da fotografia correspondente mediante confirmação. Na tela de visualização, o usuário pode percorrer as imagens armazenadas no cartão microSD e retornar à lista de registros.

A função de ajuste de data e hora permite modificar os campos referentes ao dia, mês, ano, hora e minuto. Quando o módulo de relógio de tempo real DS1307 está disponível, os valores configurados podem ser utilizados para atualizar sua referência temporal. A tela de configurações permite alterar os parâmetros previstos no firmware, incluindo a distância de detecção, o intervalo entre fotografias, o modo de iluminação infravermelha, a polaridade lógica do módulo LDR e o número de leituras consecutivas exigidas para a confirmação da presença. Também é disponibilizada a função de restauração dos parâmetros padrão. As configurações

são armazenadas em memória não volátil por meio da biblioteca Preferences, permitindo sua recuperação após a reinicialização do sistema.

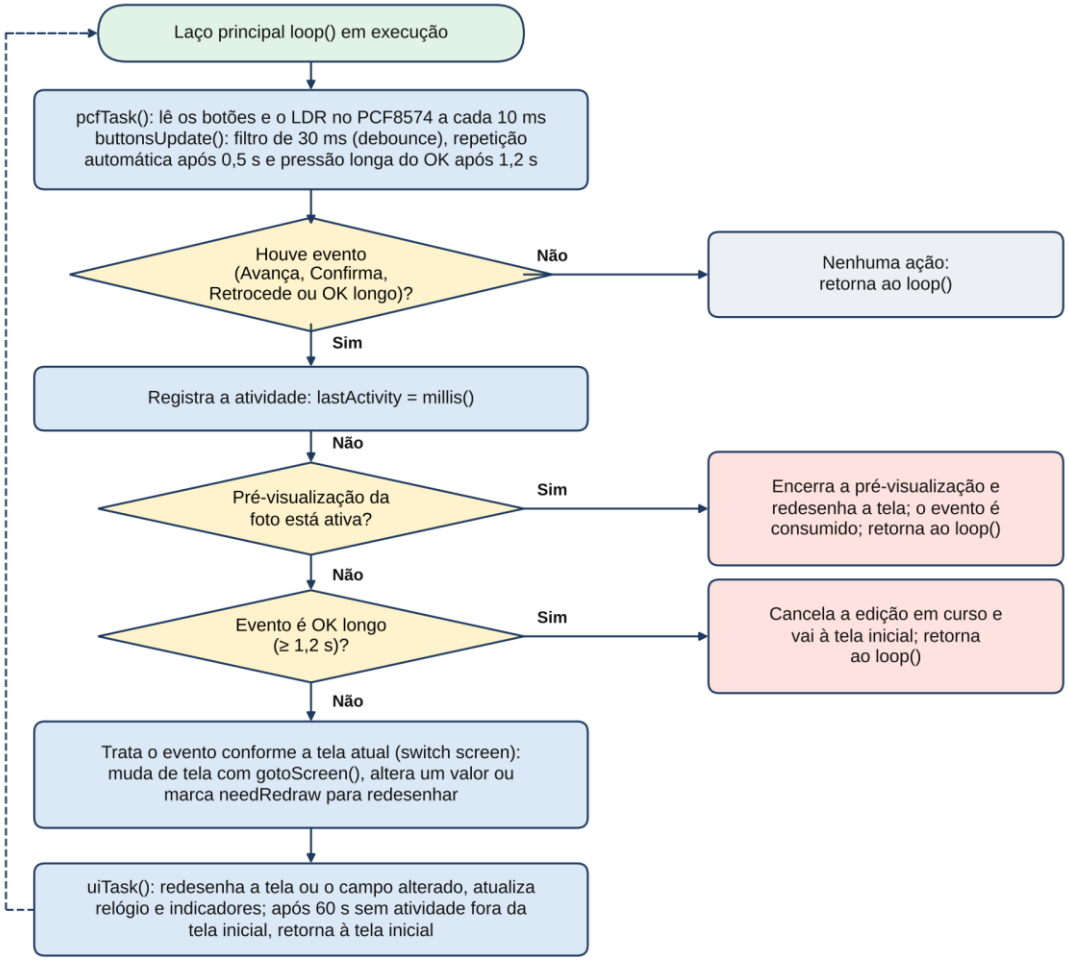
A tela de informações do sistema reúne dados de diagnóstico disponibilizados pelo firmware, como a versão do programa, o estado da câmera, a disponibilidade da memória PSRAM, a condição do relógio de tempo real, o estado do expansor PCF8574, informações do cartão microSD, a quantidade de fotografias, a memória livre e o tempo de funcionamento. A disponibilidade de cada informação depende da inicialização e do estado dos respectivos componentes ou recursos.

A navegação é realizada por três botões conectados ao PCF8574. O pino P0 está associado à função NEXT, utilizada para avançar entre opções e incrementar valores; o pino P1 corresponde à função OK, empregada para confirmar seleções e acessar o menu; e o pino P2 está associado à função BACK, utilizada para retornar entre opções ou decrementar valores durante os ajustes. Na tela principal, um acionamento curto do botão OK abre o menu principal. Nas telas secundárias, um acionamento prolongado permite retornar à tela principal. O firmware também implementa o retorno automático à tela principal após 60 segundos sem atividade nas telas secundárias.

Do ponto de vista da arquitetura de software, a atualização da interface é realizada de forma cooperativa no laço principal do programa, por meio das rotinas buttonsUpdate(), handleEvent() e uiTask(). A primeira é responsável pela atualização do estado dos botões; a segunda trata os eventos de interação; e a terceira executa as operações relacionadas à atualização visual da interface. Essas rotinas são chamadas durante a execução do loop() do Arduino. Portanto, embora o ESP32 disponha de suporte ao FreeRTOS em sua plataforma, o firmware desenvolvido para este projeto não cria tarefas independentes por meio de funções como xTaskCreate(). A organização implementada baseia-se na execução sequencial e cooperativa das rotinas do programa, sem a criação explícita de tarefas próprias para a interface.

O funcionamento da interface é descrito por dois fluxogramas, elaborados a partir da leitura do código-fonte do firmware. O primeiro (Figura 35) mostra como um evento de botão é identificado, filtrado e tratado a cada passagem pelo laço principal; o segundo (Figura 36) mostra como esses eventos conduzem o sistema entre as sete telas da interface. Como o desenvolvimento foi conduzido por meio da implementação do firmware e da avaliação lógica simulada, as figuras têm finalidade explicativa e não constituem evidência de validação física do display, dos botões ou dos demais periféricos. A sequência completa das telas renderizadas a partir do firmware é apresentada na subseção 3.8.8.

Figura 35 – Fluxograma do tratamento dos eventos dos botões pelo firmware do HydraPet Vision 2.0.



Fonte: Elaborado pelos autores (2026), com base no firmware desenvolvido.

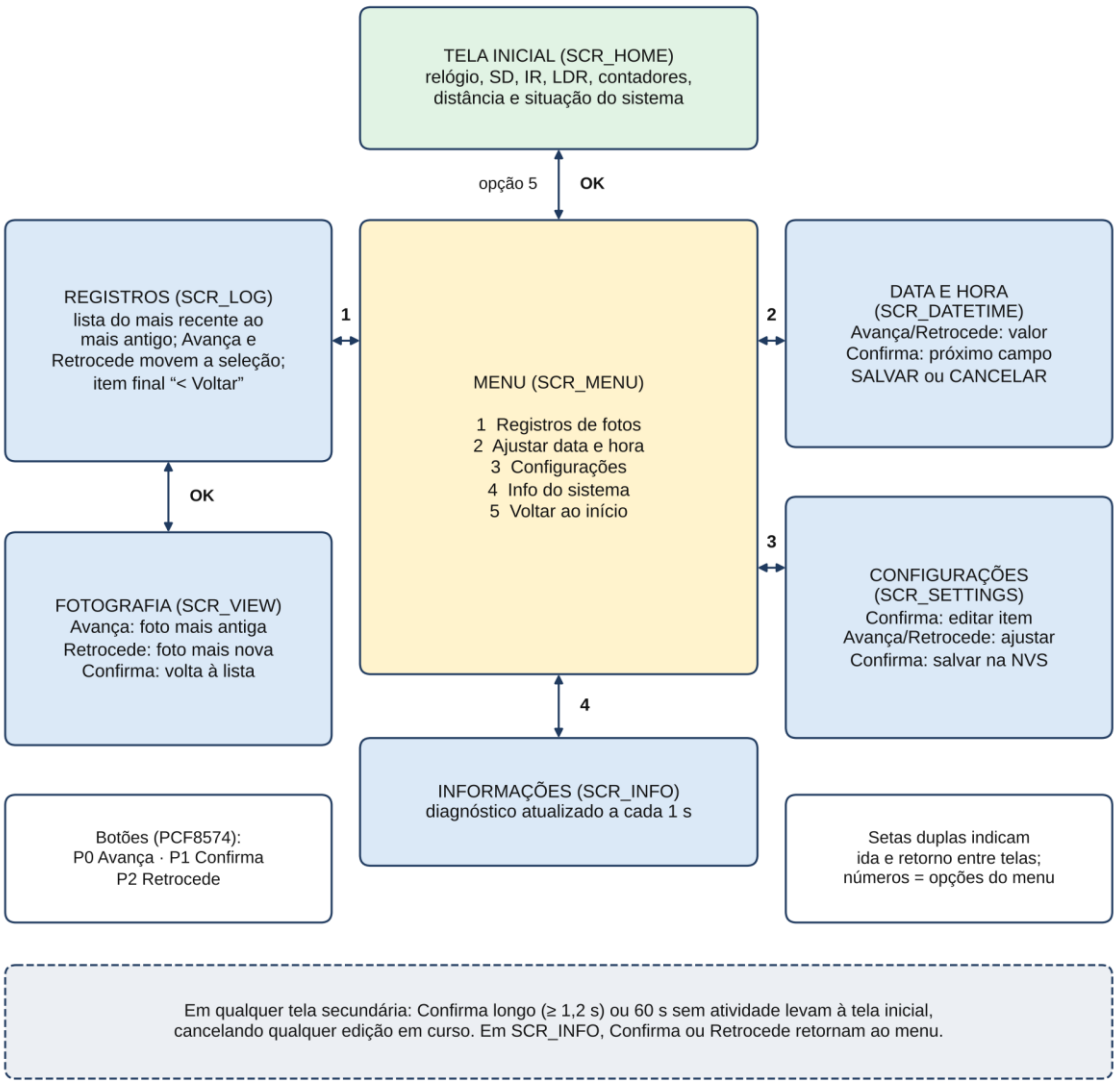
Na Figura 35, a cada ciclo do laço loop(), a rotina pcfTask() lê o expansor PCF8574, em intervalos de 10 ms, e repassa o estado dos botões à função buttonsUpdate(). Essa função aplica um filtro de 30 ms contra os ruídos de contato (debounce) e produz, no máximo, um evento por leitura: Avança, Confirma, Retrocede ou Confirma longo. Os botões Avança e Retrocede geram o evento no momento do acionamento e passam a repeti-lo a cada 120 ms depois de 0,5 s de pressão contínua, o que agiliza a alteração de valores distantes. O botão Confirma gera o evento ao ser liberado, de modo que o acionamento curto possa ser distinguido da pressão longa, reconhecida após 1,2 s.

Quando há um evento, a função handleEvent() registra a atividade do usuário, reiniciando a contagem do tempo de inatividade, e executa três verificações em sequência. A primeira trata a pré-visualização exibida após uma captura: se estiver ativa, qualquer botão apenas a encerra e redesenha a tela, sem acionar outra função. A segunda trata o Confirma

longo, que cancela uma eventual edição em curso e leva o sistema à tela inicial, a partir de qualquer tela. Somente se nenhuma das duas condições ocorrer o evento é interpretado de acordo com a tela atual, por meio de uma estrutura switch, que pode mudar de tela com gotoScreen(), alterar um valor ou apenas marcar a necessidade de redesenho. A função uiTask() executa o redesenho, atualiza o relógio e os indicadores e, em qualquer tela diferente da inicial, devolve o sistema à tela inicial após 60 s sem atividade. Como a interface é atualizada de forma cooperativa no próprio laço principal, a leitura dos botões convive com a medição de distância, o controle do infravermelho e a captura das fotografias.

A Figura 36 apresenta a navegação entre as sete telas, que correspondem aos estados do tipo enumerado Screen do firmware.

Figura 36 – Fluxograma de navegação entre as telas da interface homem-máquina do HydraPet Vision 2.0.



Fonte: Elaborado pelos autores (2026), com base no firmware desenvolvido.

A tela inicial (SCR_HOME) é o estado em que o equipamento permanece durante o monitoramento; nela, o botão Confirma abre o menu (SCR_MENU), sempre com a primeira opção selecionada. No menu, Avança e Retrocede percorrem circularmente as cinco opções, e Confirma aciona a opção destacada: Registros de fotos conduz à lista de registros (SCR_LOG); Ajustar data e hora carrega o horário atual do RTC e abre a tela de ajuste (SCR_DATETIME); Configuracoes abre a lista de parâmetros (SCR_SETTINGS); Info do sistema abre a tela de diagnóstico (SCR_INFO); e Voltar ao início retorna à tela inicial.

Na lista de registros, a seleção percorre os eventos do mais recente ao mais antigo, seguidos do item "< Voltar"; confirmar um evento abre a fotografia correspondente (SCR_VIEW), enquanto confirmar o item "< Voltar" retorna ao menu. Na tela da fotografia, Avança exibe o registro mais antigo, Retrocede exibe o mais novo e Confirma devolve o usuário à lista, mantendo a seleção no evento que acabou de ser visualizado. Na tela de data e hora, Avança e Retrocede alteram o valor do campo selecionado e Confirma passa ao campo seguinte; após o último campo, a seleção desloca-se para as ações SALVAR e CANCELAR, e qualquer uma delas, quando confirmada, retorna ao menu, sendo que a primeira grava o novo horário no RTC. Na tela de configurações, Confirma coloca o item selecionado em modo de edição, Avança e Retrocede ajustam o valor e um novo Confirma o grava na memória não volátil; os itens Restaurar padroes e Voltar executam, respectivamente, a restauração dos valores originais e o retorno ao menu. A tela de informações não possui itens editáveis: Confirma ou Retrocede retornam ao menu.

Existem ainda duas saídas gerais para a tela inicial: o acionamento longo do botão Confirma, em qualquer tela, e o esgotamento do tempo de inatividade de 60 s nas telas secundárias, o que impede que o equipamento permaneça indefinidamente em um menu. Quando uma fotografia é capturada na tela inicial, a navegação é interrompida por cerca de três segundos para a exibição da pré-visualização, que qualquer botão encerra; nas demais telas, a captura é registrada sem alterar a tela exibida.

3.8.1 Tela inicial

A tela inicial funciona como o principal painel operacional do HydraPet Vision, reunindo informações relevantes para o acompanhamento do funcionamento do protótipo. Nessa tela são apresentados o nome do sistema, a data e o horário fornecidos pelo RTC, o estado do cartão MicroSD, o modo de iluminação infravermelha e a condição atual de luminosidade. Também são exibidas a quantidade de eventos registrados no dia, o total de eventos

armazenados, a distância atualmente medida pelo sensor e mensagens relacionadas ao estado geral do sistema.

A distância do sensor ultrassônico é atualizada independentemente da navegação pelas demais funções da interface, permitindo que o usuário acompanhe continuamente a condição da região monitorada. A tela também pode apresentar mensagens relacionadas a situações de operação ou falha, incluindo problemas com a câmera, indisponibilidade do cartão MicroSD, falha de comunicação com o expansor de entradas e saídas e identificação de presença na região monitorada. Dessa maneira, a tela inicial atua não apenas como interface de operação, mas também como um painel simplificado de diagnóstico do protótipo.

3.8.2 Menu principal

O menu principal organiza as funções disponíveis no equipamento em cinco opções: Registros de fotos, Ajustar data e hora, Configurações, Informações do sistema e Voltar ao início. Essa estrutura concentra as operações de consulta, configuração e diagnóstico em uma única interface, evitando a necessidade de comandos externos para modificar os parâmetros de funcionamento. A organização também facilita a utilização do equipamento durante os ensaios e procedimentos de validação do protótipo.

A navegação entre as opções é realizada pelos botões Avança e Retrocede, enquanto o botão Confirma permite acessar a função selecionada. A lógica de navegação é integrada à máquina de estados da interface, de modo que cada opção direciona o sistema para uma tela específica. O retorno à tela inicial pode ser realizado pela própria opção correspondente do menu ou pela utilização da pressão longa do botão de confirmação, proporcionando uma forma mais rápida de abandonar uma determinada função.

3.8.3 Registros de fotos

A tela de registros permite consultar os eventos fotográficos armazenados pelo sistema, apresentando-os em ordem do mais recente para o mais antigo. Cada registro possui uma identificação numérica associada às informações temporais do evento, permitindo que o usuário reconheça rapidamente quando determinada fotografia foi realizada. Essa organização utiliza os registros mantidos em memória pelo firmware e as informações persistentes armazenadas no cartão MicroSD.

A partir da seleção de um evento, o usuário pode acessar a fotografia correspondente armazenada no diretório /fotos. Esse procedimento estabelece uma relação direta entre o

registro do evento e o arquivo de imagem produzido durante a captura. Dessa maneira, o próprio equipamento pode ser utilizado para consultar o histórico fotográfico sem que seja necessário retirar imediatamente o cartão MicroSD ou transferir os arquivos para outro computador.

3.8.4 Galeria de fotografias

A galeria de fotografias é responsável pela apresentação das imagens capturadas pelo sistema diretamente no display TFT ILI9341. Quando um evento é selecionado, a função loadFile() acessa o arquivo JPEG correspondente no cartão MicroSD e determina seu tamanho para realizar o carregamento. Quando a PSRAM está disponível na ESP32-CAM, o conteúdo da imagem é preferencialmente armazenado nessa memória, reduzindo a utilização da memória interna disponível para as demais funções do sistema.

Após o carregamento, a biblioteca TJpg_Decoder realiza a decodificação do arquivo JPEG e encaminha os dados resultantes para a função tftOutput(), responsável pela apresentação da imagem no display. A fotografia ocupa a área principal da tela, enquanto a região inferior apresenta informações relacionadas ao evento, como sua identificação, data e horário. O usuário pode utilizar os botões Avança e Retrocede para percorrer as fotografias disponíveis (Avança exibe o registro anterior, mais antigo, e Retrocede o seguinte, mais recente; Confirma retorna à lista), transformando o display do equipamento em uma pequena galeria local de registros.

3.8.5 Ajuste de data e hora

A versão 2.0 permite realizar o ajuste da data e do horário do RTC diretamente pela interface do equipamento, eliminando a necessidade de alterar essas informações diretamente no código-fonte. A tela de configuração apresenta cinco campos editáveis: dia, mês, ano, hora e minuto. O botão Avança incrementa o valor do campo selecionado, enquanto o botão Retrocede realiza a operação inversa; o botão Confirma, por sua vez, avança para o campo seguinte.

Durante o ajuste, o firmware também realiza verificações relacionadas à validade da data. A função daysInMonth() determina a quantidade de dias correspondente ao mês selecionado e considera a ocorrência de anos bissextos, evitando que sejam configuradas datas incompatíveis com o calendário. Após o preenchimento dos campos, o usuário pode confirmar o salvamento ou cancelar a operação, mantendo maior controle sobre a alteração das informações utilizadas nos registros fotográficos.

Quando o usuário confirma a alteração e o RTC está disponível, o firmware utiliza a função rtc.adjust() para gravar os valores selecionados no relógio. A atualização passa então a ser utilizada nos eventos posteriores, garantindo que novas fotografias e registros CSV recebam a referência temporal configurada. Esse recurso é particularmente importante para a rastreabilidade dos eventos, pois a data e o horário constituem informações fundamentais para a organização do histórico fotográfico.

3.8.6 Configurações do sistema

A tela de configurações representa uma das principais evoluções funcionais incorporadas à versão 2.0, permitindo adaptar o comportamento do protótipo às características do ambiente de instalação. Entre os parâmetros disponíveis estão a distância de detecção, o intervalo mínimo entre fotografias, o modo de iluminação infravermelha, a polaridade do LDR e a quantidade de leituras utilizadas para confirmação da presença. A interface também disponibiliza as opções de restauração dos padrões e retorno ao menu anterior.

A distância de detecção pode ser configurada entre 5 e 60 cm, permitindo adequar a região de monitoramento à posição do dispositivo em relação ao recipiente. O intervalo mínimo entre fotografias pode ser definido entre 3 e 120 segundos, evitando registros sucessivos durante uma mesma permanência do animal. O modo de iluminação infravermelha pode assumir as condições automático, ligado ou desligado, enquanto a polaridade do LDR pode ser invertida conforme a característica do módulo utilizado.

Outro parâmetro disponível é o filtro de presença, ajustável entre duas e dez leituras consecutivas. Esse valor determina quantas medições válidas devem ocorrer antes que a presença seja confirmada e uma captura seja solicitada. Após a confirmação das alterações, os parâmetros são armazenados na memória não volátil NVS por meio da biblioteca Preferences, permitindo que as configurações permaneçam disponíveis mesmo após o desligamento do equipamento.

A opção de restauração dos padrões permite retornar os parâmetros aos valores originalmente definidos pelo firmware, facilitando a recuperação da configuração inicial do sistema. Essa estrutura proporciona maior flexibilidade ao protótipo, pois alterações relacionadas ao ambiente de instalação podem ser realizadas diretamente pela IHM, sem necessidade de modificar ou recompilar o programa. Com isso, a versão 2.0 apresenta uma configuração mais adaptável às diferentes condições de utilização previstas para o HydraPet Vision.

3.8.7 Tela de informações do sistema

A tela de informações foi desenvolvida para apresentar uma visão geral dos principais componentes e condições operacionais do protótipo. Entre as informações disponibilizadas estão a versão do firmware, o estado da câmera, a disponibilidade da PSRAM, a condição do RTC, o estado do PCF8574 e a situação do cartão MicroSD. Também são apresentados dados relacionados à quantidade de fotografias armazenadas e às condições atuais de iluminação e acionamento do infravermelho.

A interface de diagnóstico apresenta ainda informações relacionadas à memória RAM disponível e ao tempo de funcionamento do sistema. A reunião desses dados em uma única tela facilita a identificação de condições anormais durante os ensaios e permite verificar rapidamente se os principais periféricos foram inicializados corretamente. Dessa forma, a tela de informações possui importância tanto operacional quanto experimental, contribuindo para o acompanhamento do comportamento do protótipo durante sua utilização. Nessa tela, os botões Confirma e Retrocede conduzem de volta ao menu.

3.8.8 Sequência completa das telas da interface

Com a finalidade de documentar o comportamento da interface em todos os seus estados, foram renderizadas as 45 telas que o firmware produz ao longo de um cenário completo de utilização, identificadas de T01 a T45 e reunidas em cinco painéis, apresentados nas Figuras 37 a 41. As telas reproduzem as rotinas de desenho do código-fonte, respeitando a resolução de 320 × 240 pixels, a paleta de cores definida na função initColors(), o posicionamento dos elementos gráficos e os textos efetivamente exibidos, os quais aparecem sem acentuação gráfica por limitação da fonte padrão da biblioteca Adafruit GFX.

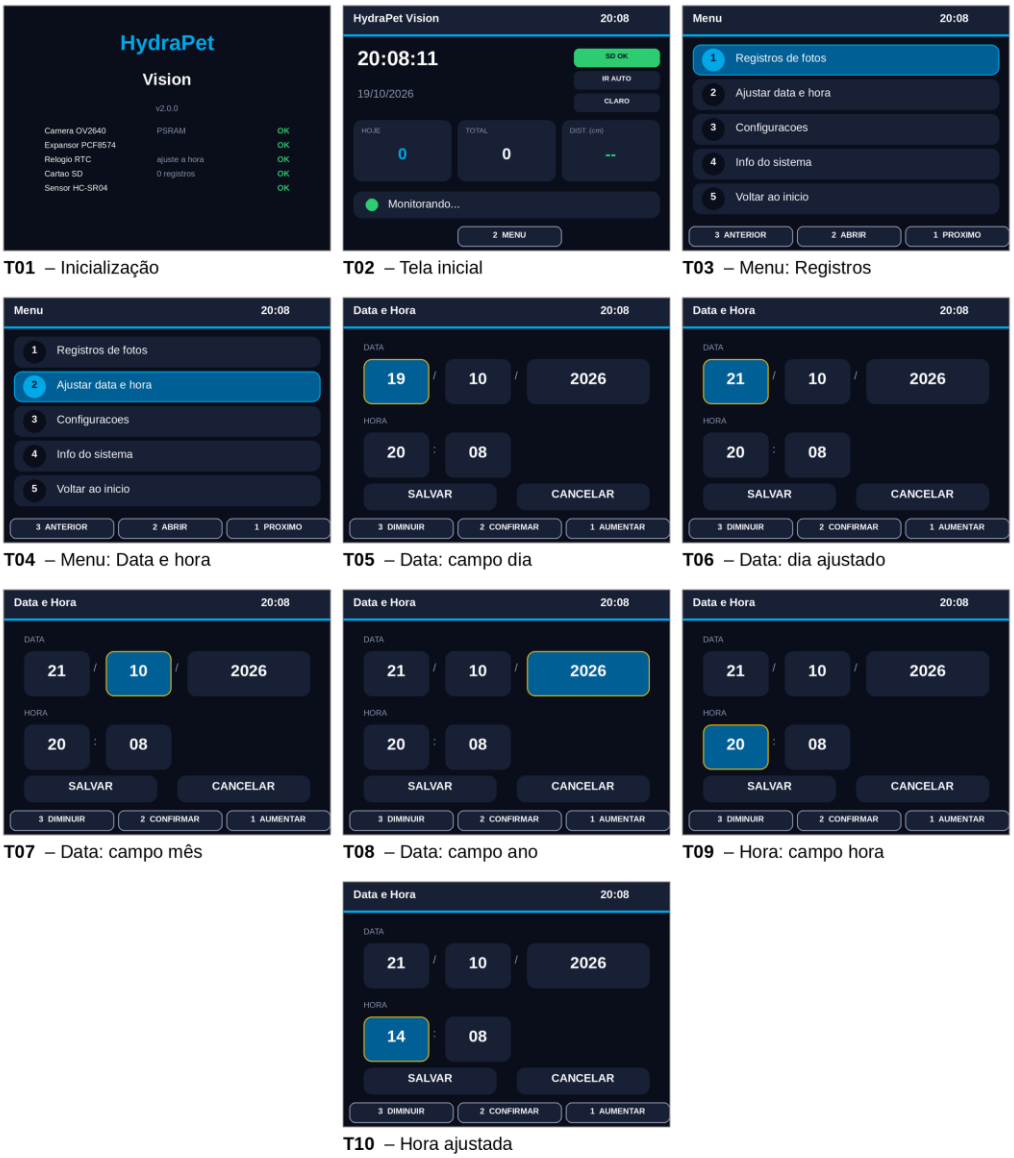
Os valores apresentados, como datas, horários, distâncias, contagens e fotografias, são ilustrativos e foram escolhidos para compor uma sequência coerente: um equipamento recém-energizado, cujo relógio ainda não foi ajustado, é configurado pelo usuário e, em seguida, submetido a duas visitas do animal ao recipiente, a primeira em condição de luz normal e a segunda em ambiente escuro, com acionamento do infravermelho. Por se tratar de renderizações, e não de capturas de um display físico, a tipografia empregada difere da fonte bitmap do firmware, sem alteração do conteúdo ou da lógica de cada tela, e a coloração esverdeada das fotografias noturnas é apenas uma convenção visual.

Como o display não é sensível ao toque, toda a navegação é realizada exclusivamente pelos três botões físicos. O rodapé de cada tela indica a função atual de cada botão: o da

esquerda, rotulado como 3, corresponde ao botão Retrocede e serve para voltar, selecionar o item anterior ou diminuir um valor; o central, rotulado como 2, corresponde ao botão Confirma e serve para abrir, editar, confirmar ou salvar; e o da direita, rotulado como 1, corresponde ao botão Avança e serve para selecionar o próximo item ou aumentar um valor. A numeração de T01 a T45 segue a ordem em que as telas surgem durante o cenário descrito.

O primeiro painel (Figura 37) reúne as telas T01 a T10, que correspondem à inicialização do equipamento, à tela inicial, ao menu principal e ao início do ajuste de data e hora.

Figura 37 – Telas T01 a T10 da IHM: inicialização, tela inicial, menu e início do ajuste de data e hora.



T01 – Inicialização
T02 – Tela inicial
T03 – Menu: Registros
T04 – Menu: Data e hora
T05 – Data: campo dia
T06 – Data: dia ajustado
T07 – Data: campo mês
T08 – Data: campo ano
T09 – Hora: campo hora
T10 – Hora ajustada

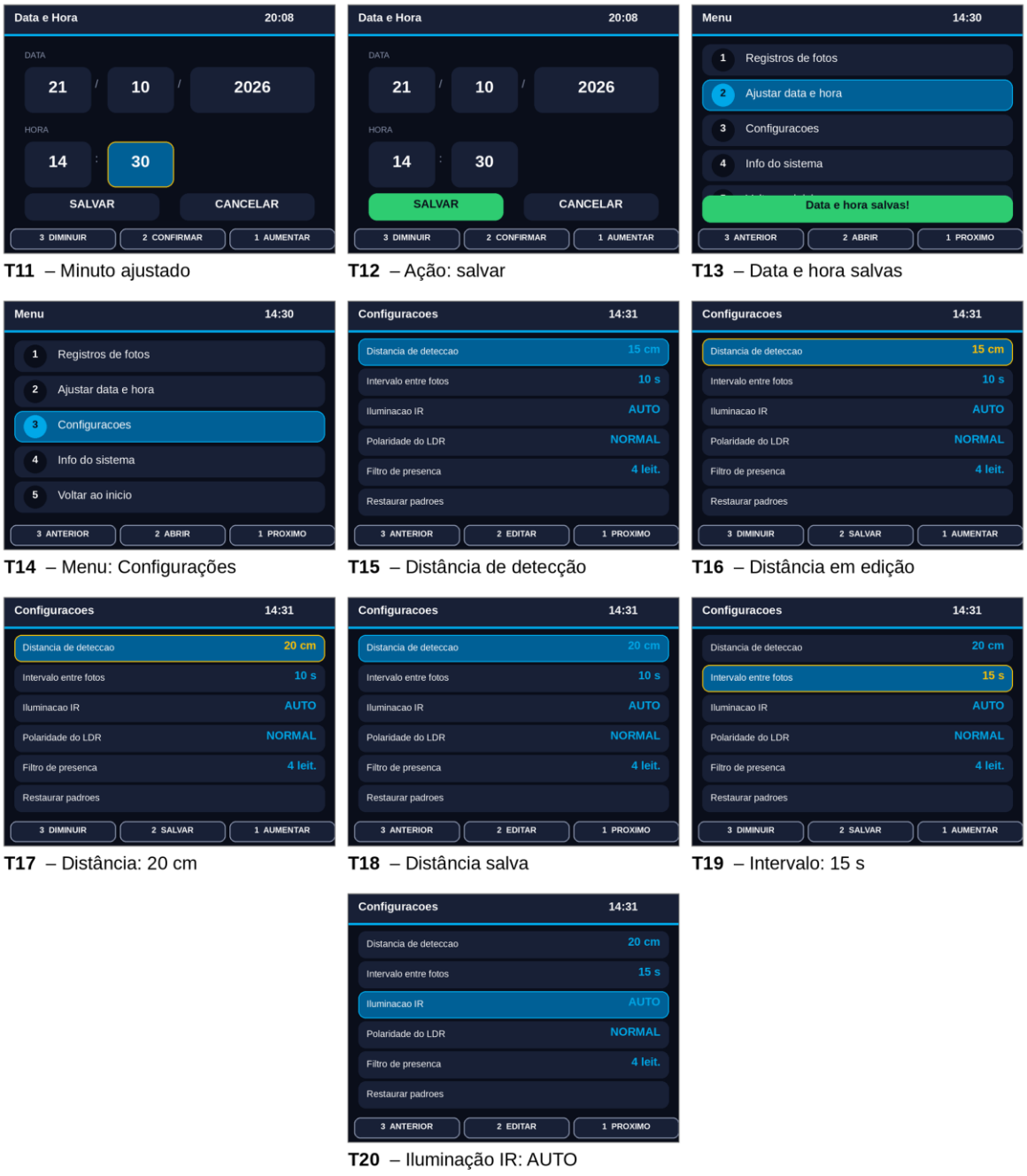
Fonte: Elaborado pelos autores (2026), a partir das rotinas de desenho do firmware HydraPet Vision v2.0 (dados e fotografias ilustrativos).

A tela T01 é exibida durante a execução da função setup(). Após a apresentação do nome do sistema e da versão do firmware, o programa lista o resultado da inicialização da câmera OV2640, com a indicação de memória PSRAM, do expansor PCF8574, do relógio RTC, do cartão MicroSD, com a quantidade de registros lidos do arquivo CSV, e do sensor HC-SR04, cujo estado depende da resposta do expansor responsável por seu acionamento. Cada linha é acompanhada da indicação OK ou ERRO. No cenário simulado, o relógio informa "ajuste a hora", condição sinalizada pelo firmware quando o RTC estava parado e foi reinicializado com a data e a hora da compilação do programa. Após cerca de 1,2 segundo, o sistema passa à tela principal (T02), que exibe o relógio com segundos, a data, os indicadores de cartão (SD OK), de modo de iluminação (IR AUTO) e de luminosidade (CLARO), os cartões de contagem HOJE e TOTAL, a distância medida pelo sensor, apresentada como "--" quando não há leitura válida, e a mensagem de estado "Monitorando...". O rodapé dessa tela possui apenas o botão central, rotulado como MENU.

Ao acionar o botão Confirma na tela principal, abre-se o menu (T03), cujo primeiro item, Registros de fotos, vem selecionado. O menu possui cinco opções numeradas (Registros de fotos, Ajustar data e hora, Configuracoes, Info do sistema e Voltar ao inicio), e o item selecionado é destacado em azul. O botão Avança desloca a seleção para a opção seguinte, como ocorre em T04, em que se seleciona Ajustar data e hora, e a navegação é circular, de modo que, após a última opção, retorna-se à primeira. Ao confirmar essa opção, o firmware lê o horário atual do RTC e apresenta a tela de ajuste (T05), com os campos de dia, mês e ano na parte superior, os de hora e minuto abaixo e as ações SALVAR e CANCELAR. O campo selecionado recebe contorno amarelo, e o rodapé passa a indicar as funções DIMINUIR, CONFIRMAR e AUMENTAR. Em T06, o dia é alterado de 19 para 21 por dois acionamentos do botão Avança. A partir daí, o botão Confirma avança de campo, passando pelo mês (T07), pelo ano (T08) e pela hora (T09), cujos limites são, respectivamente, de 1 a 12, de 2024 a 2099 e de 0 a 23, enquanto o limite do dia depende da quantidade de dias do mês, calculada pela função daysInMonth(). Em T10, a hora é reduzida de 20 para 14 por seis acionamentos do botão Retrocede.

O segundo painel (Figura 38) apresenta as telas T11 a T20, que concluem o ajuste de data e hora e iniciam a configuração dos parâmetros de funcionamento.

Figura 38 – Telas T11 a T20 da IHM: conclusão do ajuste de data e hora e início da configuração dos parâmetros.



T11 – Minuto ajustado
T12 – Ação: salvar
T13 – Data e hora salvas
T14 – Menu: Configurações
T15 – Distância de detecção
T16 – Distância em edição
T17 – Distância: 20 cm
T18 – Distância salva
T19 – Intervalo: 15 s
T20 – Iluminação IR: AUTO

Fonte: Elaborado pelos autores (2026), a partir das rotinas de desenho do firmware HydraPet Vision v2.0 (dados e fotografias ilustrativos).

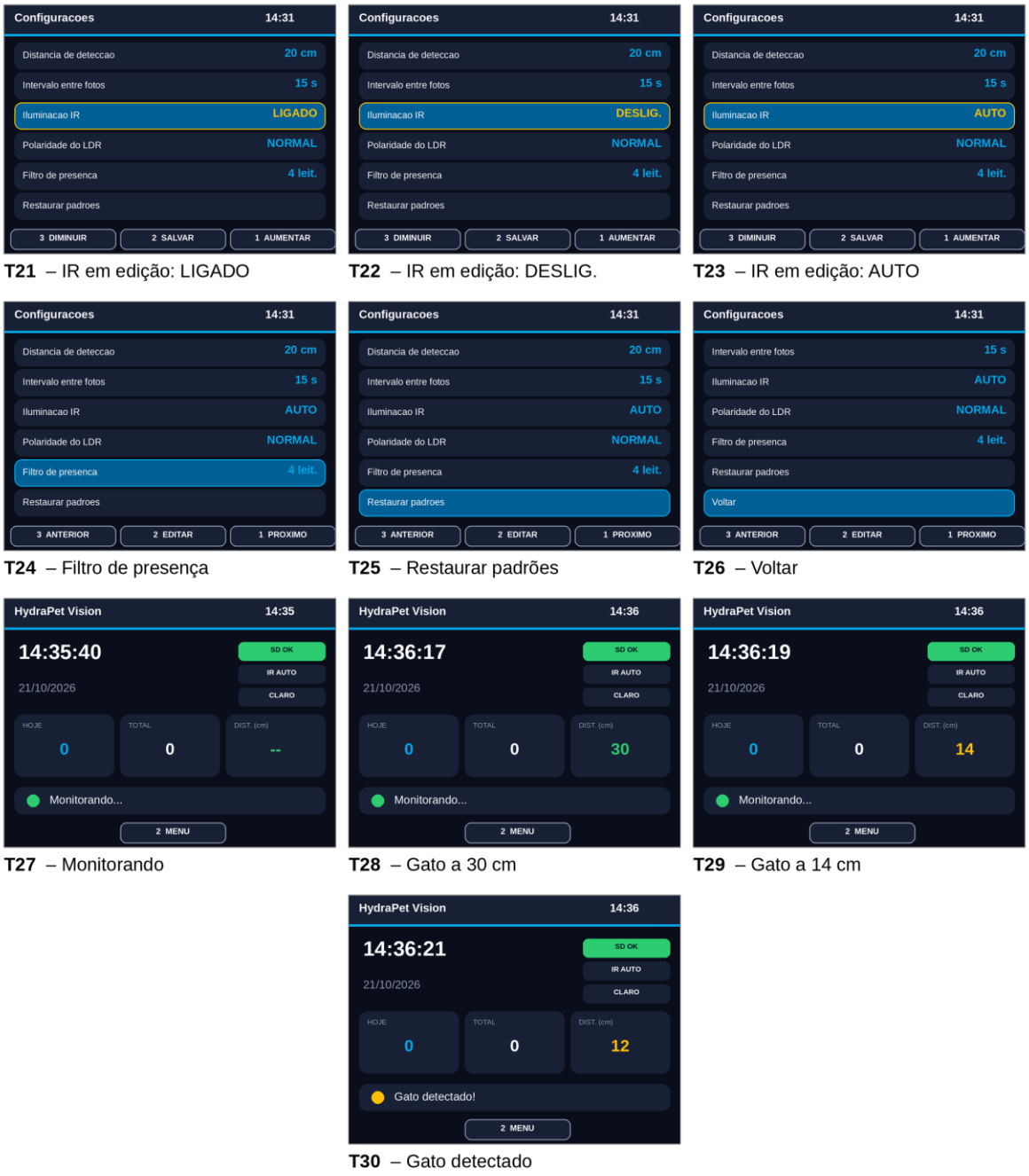
Em T11, o campo de minutos é ajustado de 08 para 30; como os botões laterais possuem repetição automática após 500 ms de pressão contínua, valores distantes podem ser alcançados mantendo-se o botão pressionado. Ao confirmar o último campo, a seleção passa à linha de ações, e a opção SALVAR é destacada em verde (T12). Acionando-se novamente o botão

Confirma, o firmware grava os valores no RTC por meio da função rtc.adjust(), com segundos zerados, e retorna ao menu, onde exibe por cerca de 2,5 segundos a mensagem “Data e hora salvas!” (T13). Nessa tela, o relógio do cabeçalho já mostra o novo horário, 14:30. Caso a opção CANCELAR fosse selecionada pelos botões Avança ou Retrocede e confirmada, o menu seria exibido sem alteração do relógio e, se o RTC não fosse encontrado, a mensagem de erro “RTC nao encontrado” substituiria a de sucesso.

As telas seguintes tratam da configuração. Em T14, o botão Avança seleciona a opção Configuracoes, e a sua confirmação abre a lista de parâmetros (T15), composta por sete itens, dos quais seis são visíveis simultaneamente: Distancia de deteccao, Intervalo entre fotos, Iluminacao IR, Polaridade do LDR, Filtro de presenca, Restaurar padroes e Voltar. Os valores exibidos em T15 (15 cm, 10 s, AUTO, NORMAL e 4 leituras) são os padrões definidos pela função cfgDefaults(). Ao confirmar um item, ele entra em modo de edição: o contorno e o valor passam à cor amarela e o rodapé muda para DIMINUIR, SALVAR e AUMENTAR (T16). Em T17, a distância de detecção é elevada para 20 cm, valor que pode variar de 5 a 60 cm e que, ao ser confirmado, é gravado na memória não volátil, de modo que a lista retorna ao modo de navegação (T18). Em T19, o item Intervalo entre fotos, ajustável de 3 a 120 segundos, é editado de 10 para 15 s e, depois de salvo, a seleção segue para a Iluminacao IR, ainda no modo AUTO (T20).

O terceiro painel (Figura 39) contém as telas T21 a T30, que completam a configuração e iniciam a simulação de uma visita do animal em condição de luz normal.

Figura 39 – Telas T21 a T30 da IHM: modos do infravermelho, demais parâmetros e aproximação do animal.



T21 – IR em edição: LIGADO

T22 – IR em edição: DESLIG.

T23 – IR em edição: AUTO

T24 – Filtro de presença

T25 – Restaurar padrões

T26 – Voltar

T27 – Monitorando

T28 – Gato a 30 cm

T29 – Gato a 14 cm

T30 – Gato detectado

Fonte: Elaborado pelos autores (2026), a partir das rotinas de desenho do firmware HydraPet Vision v2.0 (dados e fotografias ilustrativos).

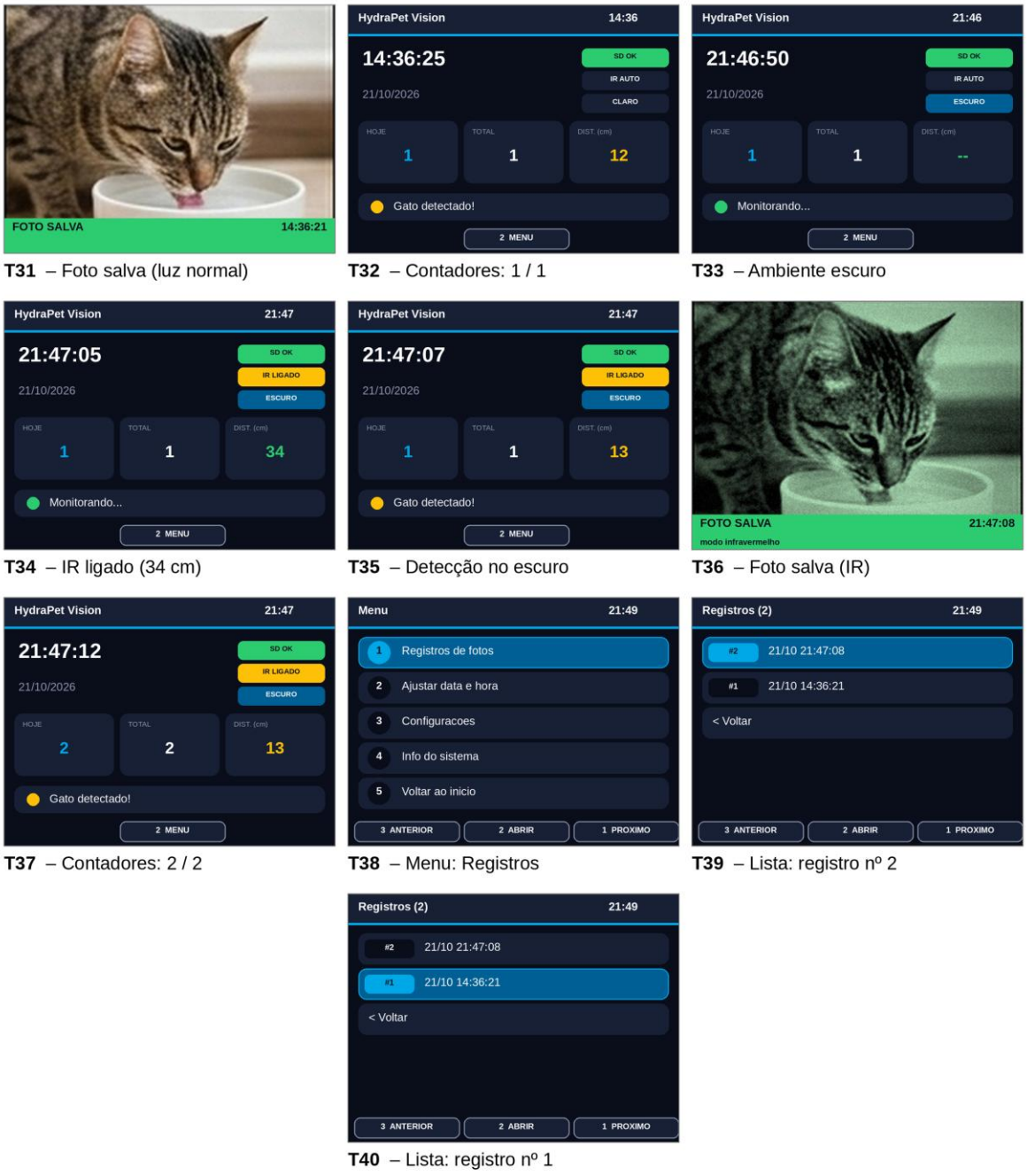
Em T21, T22 e T23, o modo de iluminação infravermelha é percorrido pelo botão Avança em ordem cíclica: de AUTO para LIGADO, em que os LEDs permanecem sempre acesos; depois para DESLIG., em que permanecem sempre apagados; e novamente para AUTO, em que são acionados apenas em ambientes escuros e na aproximação do animal. Prosseguindo

na lista, o item Polaridade do LDR alterna entre NORMAL e INVERTIDA, de acordo com o sinal digital do módulo de luminosidade, e o Filtro de presenca, mostrado em T24 com o valor de 4 leituras, define quantas medições consecutivas dentro do limite são necessárias para validar a presença, entre 2 e 10. O item Restaurar padroes, selecionado em T25, devolve todos os parâmetros aos valores originais e exibe a mensagem “Padroes restaurados” quando confirmado. Por fim, o item Voltar (T26), alcançado com a rolagem da lista, retorna ao menu quando confirmado.

Concluída a configuração, o equipamento retorna à tela principal, seja pela opção Voltar ao inicio do menu, seja por uma pressão longa do botão Confirma, de cerca de 1,2 segundo, que leva à tela inicial a partir de qualquer tela secundária. Em T27, o sistema monitora o ambiente sem leitura de distância válida. Quando o animal se aproxima, o sensor ultrassônico passa a fornecer medições: em T28, a distância de 30 cm, superior ao limite de 20 cm configurado, é exibida em verde e a mensagem permanece “Monitorando...”; em T29, a distância de 14 cm, inferior ao limite, passa a ser exibida em amarelo, mas a presença ainda não é confirmada, pois o firmware exige a quantidade configurada de leituras consecutivas dentro do limite. Somente quando esse critério é satisfeito a mensagem muda para “Gato detectado!” (T30), com indicador amarelo, e a captura é solicitada, desde que tenha transcorrido o intervalo mínimo desde a última fotografia.

O quarto painel (Figura 40) reúne as telas T31 a T40, que mostram o resultado das capturas em luz normal e em ambiente escuro, a atualização dos contadores e o início da consulta aos registros.

Figura 40 – Telas T31 a T40 da IHM: capturas com luz normal e infravermelha, contadores e lista de registros.



Fonte: Elaborado pelos autores (2026), a partir das rotinas de desenho do firmware HydraPet Vision v2.0 (dados e fotografias ilustrativos).

Realizada a captura, a câmera fornece um quadro JPEG em resolução QVGA, que é gravado no cartão MicroSD e exibido por três segundos em tela cheia, com uma faixa verde contendo a mensagem “FOTO SALVA” e o horário do registro (T31). O acionamento de qualquer botão durante esse intervalo encerra a pré-visualização. Em seguida, a tela principal é

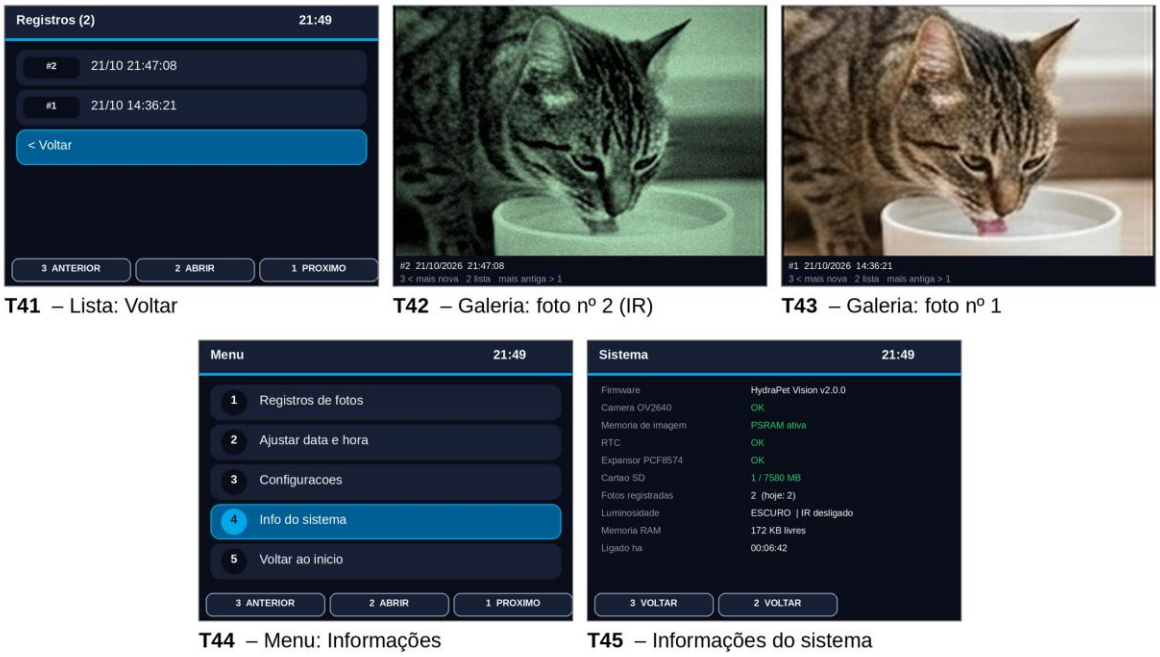
redesenhada com os contadores atualizados, HOJE e TOTAL iguais a 1 (T32); como o animal permanece dentro do limite, a mensagem “Gato detectado!” é mantida, mas uma nova fotografia só será solicitada após o intervalo configurado, de 15 segundos, o que evita registros repetidos de uma mesma visita.

As telas T33 a T37 simulam a visita em ambiente escuro. Em T33, o módulo LDR indica baixa luminosidade e, depois de a condição permanecer estável por três segundos, o indicador passa de CLARO para ESCURO, enquanto o modo IR AUTO mantém os LEDs apagados. Em T34, o animal é detectado a 34 cm, distância superior ao limite de 20 cm, mas inferior ao seu dobro; nessa faixa, o firmware antecipa o acionamento do infravermelho, e o indicador passa a IR LIGADO, em amarelo. Em T35, a distância de 13 cm e a confirmação por leituras consecutivas resultam na mensagem “Gato detectado!” e, em T36, a fotografia noturna é exibida com a faixa verde de confirmação, o horário e a indicação “modo infravermelho”. A tela principal volta a ser exibida em T37, com os contadores HOJE e TOTAL iguais a 2 e o indicador de infravermelho ainda ativo, pois os LEDs são mantidos acesos por um breve intervalo após a captura.

A consulta aos registros inicia-se no menu, em que a opção Registros de fotos (T38) é confirmada. A tela resultante (T39) apresenta, no cabeçalho, a quantidade total de registros e, na lista, os eventos do mais recente para o mais antigo, cada um com seu número, a data e o horário. Em T39, o registro nº 2 está selecionado e, em T40, o botão Avança desloca a seleção para o registro nº 1.

O quinto painel (Figura 41) apresenta as telas T41 a T45, que completam a consulta aos registros e a verificação do estado do sistema.

Figura 41 – Telas T41 a T45 da IHM: lista de registros, galeria de fotografias e informações do sistema.



T41 – Lista: Voltar **T42** – Galeria: foto nº 2 (IR) **T43** – Galeria: foto nº 1

T44 – Menu: Informações **T45** – Informações do sistema

Fonte: Elaborado pelos autores (2026), a partir das rotinas de desenho do firmware HydraPet Vision v2.0 (dados e fotografias ilustrativos).

A lista de registros é circular e inclui, ao final, a opção < Voltar, selecionada em T41, que, ao ser confirmada, retorna ao menu. Quando o botão Confirma é acionado sobre um registro, a função drawView() lê o arquivo JPEG correspondente no cartão MicroSD e o exibe em tela cheia, com uma faixa inferior que informa o número do registro, a data e o horário, além de lembrar a função dos botões: Retrocede para a foto mais nova, Confirma para voltar à lista e Avança para a mais antiga. A tela T42 mostra o registro nº 2, capturado com infravermelho, e a T43, após um acionamento do botão Avança, mostra o registro nº 1, capturado com luz normal. Se o arquivo não for localizado, a tela exibe a mensagem “Foto nao encontrada”.

Por fim, a opção Info do sistema do menu (T44), quando confirmada, abre a tela de diagnóstico (T45). Nela são listados a versão do firmware, o estado da câmera OV2640, da memória de imagem (PSRAM), do RTC e do expansor PCF8574, a ocupação do cartão MicroSD em megabytes, a quantidade de fotografias registradas e a do dia, a condição de luminosidade com o estado do infravermelho, a memória RAM livre e o tempo de funcionamento desde a energização. Essas informações são atualizadas a cada segundo, e os valores de capacidade do cartão, de memória e de tempo de funcionamento exibidos na figura

são apenas ilustrativos, pois dependem do cartão instalado e do instante da consulta. Os botões Retrocede e Confirma retornam ao menu.

Em síntese, as 45 telas distribuem-se entre os sete estados lógicos de tela previstos no firmware, que somam 42 telas, a saber: tela principal, em seus diferentes estados (10 telas); menu principal (6); ajuste de data e hora (8); configurações (12); lista de registros (3); galeria de fotografias (2); e informações do sistema (1). A elas somam-se a tela de abertura (1) e as duas pré-visualizações exibidas após as capturas (2). Em qualquer tela secundária, a inatividade por 60 segundos provoca o retorno automático à tela principal, e a pressão longa do botão Confirma realiza o mesmo retorno de forma imediata. Reitera-se que as imagens são renderizações derivadas da lógica do firmware e não constituem registros de um equipamento fisicamente montado nem evidência de validação experimental do display e dos botões, conforme delimitado na subseção 4.1.1.

3.9 Bibliotecas e Funções Principais do Firmware

A estrutura de software do HydraPet Vision 2.0 utiliza bibliotecas específicas para cada grupo de recursos empregados no protótipo, permitindo organizar o firmware de acordo com as responsabilidades de seus periféricos. Essa divisão reduz a complexidade da implementação e facilita a manutenção do código, uma vez que as funções relacionadas à interface gráfica, comunicação, câmera, armazenamento e relógio são tratadas por componentes específicos. A integração entre essas bibliotecas é realizada pelo firmware desenvolvido para a ESP32-CAM.

Para o desenvolvimento da interface gráfica, a biblioteca Adafruit GFX fornece recursos básicos para desenho de textos, formas geométricas, retângulos, círculos e demais elementos utilizados nas telas. A biblioteca Adafruit ILI9341, por sua vez, é responsável pela comunicação específica com o controlador do display TFT. Essa combinação permite estruturar as diferentes telas da IHM e apresentar ao usuário as informações provenientes dos sensores, registros e configurações do sistema.

A comunicação com o relógio de tempo real é realizada com o auxílio da biblioteca RTClib, utilizada para leitura e ajuste do DS1307. Para a apresentação das fotografias, a biblioteca TJpg_Decoder realiza a decodificação das imagens JPEG armazenadas no cartão MicroSD. Já a biblioteca Preferences fornece acesso à memória não volátil NVS utilizada para preservar as configurações do protótipo, como distância de detecção, intervalo entre fotografias, modo de iluminação, polaridade do LDR e filtro de presença.

O gerenciamento dos arquivos e do cartão MicroSD utiliza as bibliotecas FS e SD_MMC, responsáveis pela interação com o sistema de arquivos e pelo armazenamento persistente das fotografias e registros CSV. A biblioteca esp_camera realiza o controle da câmera OV2640, incluindo a aquisição das imagens e o gerenciamento dos framebuffers utilizados durante o processo de captura. Essas funções são fundamentais para integrar o registro fotográfico ao restante da arquitetura do dispositivo.

Por fim, a biblioteca Wire fornece a comunicação pelo barramento I2C utilizado pelo RTC e pelo expansor PCF8574, além de compartilhar o barramento com a interface de controle da câmera. A utilização dessas bibliotecas permite dividir o firmware em funções específicas, mantendo uma relação clara entre software e hardware. Essa organização contribui para a manutenção do sistema e cria uma base adequada para futuras ampliações do HydraPet Vision, como novos recursos de processamento, comunicação e análise dos registros.

3.10 Máquina de Estados e Organização do Fluxo Operacional

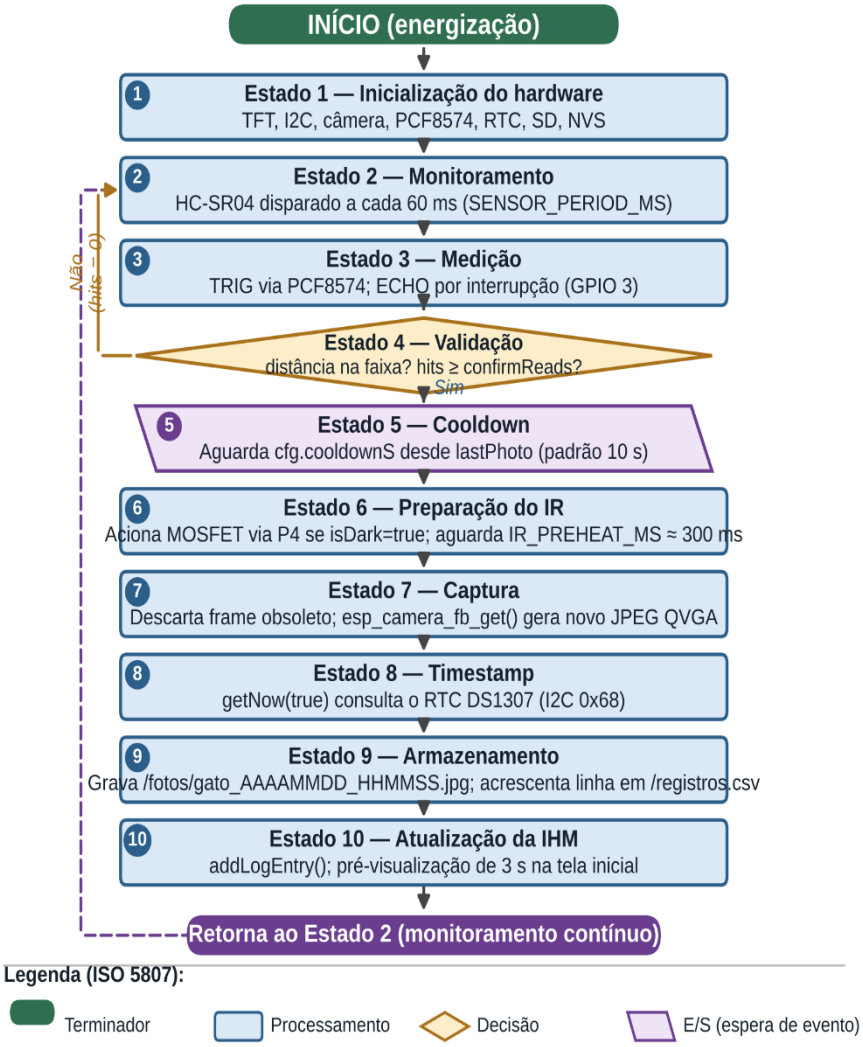
Embora o firmware não utilize uma classe formal de máquina de estados, sua organização apresenta estados funcionais bem definidos e encadeados. A operação do protótipo pode ser representada pelas seguintes condições: no Estado 1 ocorre a Inicialização, onde o sistema configura o hardware, a câmera, o RTC, o PCF8574, o cartão MicroSD, o display e os parâmetros persistentes; no Estado 2 é feito o Monitoramento, no qual o HC-SR04 é acionado periodicamente enquanto o sistema aguarda aproximações; no Estado 3 executa-se a Medição, com o pino TRIG sendo acionado para aguardar o retorno do ECHO por interrupção.

Dando continuidade ao ciclo lógico de processamento, no Estado 4 realiza-se a Validação, momento em que a distância é comparada ao limite e o contador de leituras consecutivas é atualizado; no Estado 5 entra-se em Cooldown, aguardando o intervalo configurado após uma captura; no Estado 6 faz-se a Preparação do IR, ativando a iluminação infravermelha se necessário; no Estado 7 ocorre a Captura com a geração do framebuffer JPEG; no Estado 8 obtém-se o Timestamp pelo RTC; no Estado 9 efetua-se o Armazenamento da imagem e dos dados no CSV; e no Estado 10 executa-se a Atualização da IHM.

A representação gráfica desse processo sequencial pode ser visualizada no fluxograma detalhado do capítulo. A principal diferença em relação ao fluxo anterior é que a versão 2.0 não depende de uma sequência rígida de espera para realizar o sensoriamento. O programa verifica continuamente as condições de cada subsistema e utiliza marcadores temporais para determinar quando uma tarefa deve ser executada. Essa abordagem é particularmente importante em um

sistema que precisa lidar simultaneamente com sensor, câmera, cartão SD, display, RTC, iluminação e interface gráfica.

Figura 42 – Fluxo lógico de funcionamento do firmware HydraPet Vision v2.0.



Fonte: Elaborado pelos autores (2026).

3.11 Tratamento de Falhas e Recuperação do Sistema

A confiabilidade do protótipo não depende somente do seu funcionamento sob condições ideais. Por essa razão, a versão 2.0 incorpora mecanismos avançados de diagnóstico e recuperação em tempo de execução. Durante a inicialização, o sistema registra separadamente o estado de prontidão da câmera, do RTC, do PCF8574 e do cartão MicroSD. Caso o cartão não

esteja disponível, o sistema continua funcionando em modo degradado; contudo, as fotografias não são gravadas enquanto a memória permanecer inacessível para gravação.

Para solucionar essa limitação, o firmware possui uma tarefa dedicada denominada sdRetryTask(), responsável por verificar periodicamente a presença do armazenamento. Essa tentativa de reconexão ocorre automaticamente a cada 15 segundos e apenas fora de uma visita, pois a remontagem do cartão pode bloquear o laço por centenas de milissegundos (revisão 2.1). Quando a montagem é restabelecida, o sistema recarrega os registros do arquivo CSV e atualiza as informações da capacidade do cartão. Essa característica é crucial em equipamentos embarcados, pois evita que falhas momentâneas exijam a reinicialização manual do hardware.

O mesmo princípio de tolerância a falhas é aplicado na comunicação com o expansor de I/O PCF8574. Caso o CI deixe de responder no barramento I2C, o firmware sinaliza a variável pcfOk como falsa e inicia novas tentativas de comunicação, informando a condição diretamente na IHM. O tratamento de erros também ocorre durante a captura de imagem: se a câmera não fornecer um framebuffer válido, o disparo é cancelado; se a gravação no cartão falhar, o SD é marcado como indisponível. Tais rotinas tornam o software mais previsível e seguro.

3.12 Estratégias de Temporização e Redução de Bloqueios

A gestão temporal constitui um dos principais pontos de evolução arquitetural entre o firmware original e a versão 2.0. Na primeira implementação, a leitura do sensor ultrassônico utilizava a função bloqueante pulseIn(), a qual travava a execução do processador enquanto aguardava o término do pulso de retorno. Na versão 2.0, a medição do sinal ECHO é realizada integralmente por interrupção externa. Além disso, as tarefas principais utilizam a função millis() para viabilizar um controle temporal assíncrono e eficiente.

Dentro dessa estrutura cooperativa, o sensor ultrassônico é programado para disparar novas medições a cada 60 ms, enquanto o módulo LDR exige 3 segundos de estabilidade antes de confirmar qualquer mudança na iluminação. Os botões contam com um tempo de debounce de 30 ms, e a pressão longa exige 1.200 ms para ativação. A repetição da navegação pelos botões inicia-se após 500 ms, ocorrendo em intervalos de 120 ms. Por sua vez, a janela de pré-aquecimento do infravermelho permanece ativa por até 2,5 segundos antes do evento visual.

Complementando os temporizadores do sistema, a iluminação adicional durante o disparo utiliza cerca de 300 ms de pré-aquecimento quando requisitada. A interface do menu retorna automaticamente à tela inicial após 60 segundos de inatividade do usuário, e o cartão

SD defeituoso é retestado a cada 15 segundos. A aplicação sistemática de temporizadores permite ao firmware gerenciar múltiplas funções com bloqueios curtos e conhecidos. O pequeno delay(1) ao final do loop atua apenas como pausa mínima, não interferindo nos ciclos.

3.13 Fluxo Completo de Funcionamento do HydraPet Vision 2.0

O funcionamento do sistema pode ser descrito de forma sequencial a partir da energização do protótipo. Inicialmente, o ESP32-CAM configura o display e exibe a tela de apresentação do projeto. Em seguida, as configurações operacionais armazenadas na memória NVS são recuperadas, aplicando-se os valores padrão caso não existam dados. Na sequência, o barramento I2C é configurado, a câmera é inicializada, o expansor PCF8574 é testado e o módulo RTC recupera a referência temporal para o sistema embarcado.

Após a montagem do cartão MicroSD e leitura do arquivo CSV, o sensor HC-SR04 passa a ser monitorado via interrupção e o LDR realiza a leitura inicial, exibindo a tela principal. A partir desse instante, o sistema entra em monitoramento contínuo: o ultrassom efetua disparos periódicos e, ao receber o retorno do ECHO, a interrupção calcula a duração do pulso. A distância é convertida para centímetros e, caso esteja fora da faixa programada, o contador de confirmação é zerado; caso contrário, o contador é incrementado.

Quando a quantidade de leituras consecutivas atinge o valor limite, o firmware valida a presença do animal. O sistema então verifica a luminosidade e ativa os LEDs infravermelhos se o ambiente estiver escuro e o modo automático selecionado. Respeitado o intervalo de cooldown entre fotos, a captura é solicitada: a câmera descarta o quadro anterior e gera um novo buffer JPEG. O RTC fornece a data e a hora para a nomeação do arquivo e gravação da linha de metadados no arquivo registros.csv.

Concluído o salvamento da imagem no subdiretório /fotos/AAAA-MM-DD, o evento é inserido no histórico em memória RAM e os contadores da interface são devidamente atualizados. Se o sistema estiver exibindo a tela inicial no momento do disparo, a fotografia capturada é apresentada temporariamente no display TFT como confirmação visual. Após o término dessa pré-visualização, o sistema retorna de forma automática ao modo de monitoramento. Esse ciclo de operação permanece ativo continuamente enquanto o equipamento estiver alimentado.

3.14 Integração Entre Hardware e Software

Um dos aspectos mais relevantes no desenvolvimento da versão 2.0 é a estreita integração entre as escolhas de hardware e software. A adoção do expansor PCF8574 não foi apenas uma modificação física no circuito, pois exigiu que o firmware passasse a manipular portas de entrada e saída por comunicação I2C. Da mesma forma, a substituição da leitura bloqueante por interrupção no sinal ECHO demandou a criação de um conjunto de variáveis de controle como echoRise, echoWidth, echoDone e waitingEcho.

A adição do sensor de luz LDR exigiu uma lógica dedicada para estabilização temporal, representada pelas variáveis ldrPending, ldrSince e isDark. O controle da matriz de iluminação infravermelha passou a ser gerenciado pelas variáveis de estado irState e irHoldUntil. Por sua vez, a expansão da interface gráfica levou à criação de uma enumeração de telas (SCR_HOME, SCR_MENU, SCR_LOG, SCR_VIEW, SCR_DATETIME, SCR_SETTINGS e SCR_INFO) para organizar a navegação do usuário.

A persistência de parâmetros e o histórico de eventos demandaram a criação das estruturas de dados Config e LogEntry. Dessa forma, constata-se que a evolução do projeto não se limitou ao acréscimo de componentes eletrônicos, uma vez que cada recurso de hardware foi acompanhado por uma camada correspondente de software. Essa integração harmoniosa é uma característica fundamental em engenharia de sistemas embarcados, onde o circuito físico e o firmware devem atuar como uma estrutura única.

3.15 Considerações Sobre a Montagem e Programação do Protótipo

A programação do ESP32-CAM exige cuidados específicos devido ao compartilhamento de multiplexação de suas GPIOs. A GPIO 0 é utilizada pela câmera como sinal de clock (XCLK) e também participa do processo de gravação do código. Para gravar o firmware, é necessário conectar temporariamente a GPIO 0 ao GND e, após o término do carregamento, o jumper deve ser removido para que a placa reinicie no modo de execução normal. Esse procedimento é indispensável para o correto funcionamento do sistema.

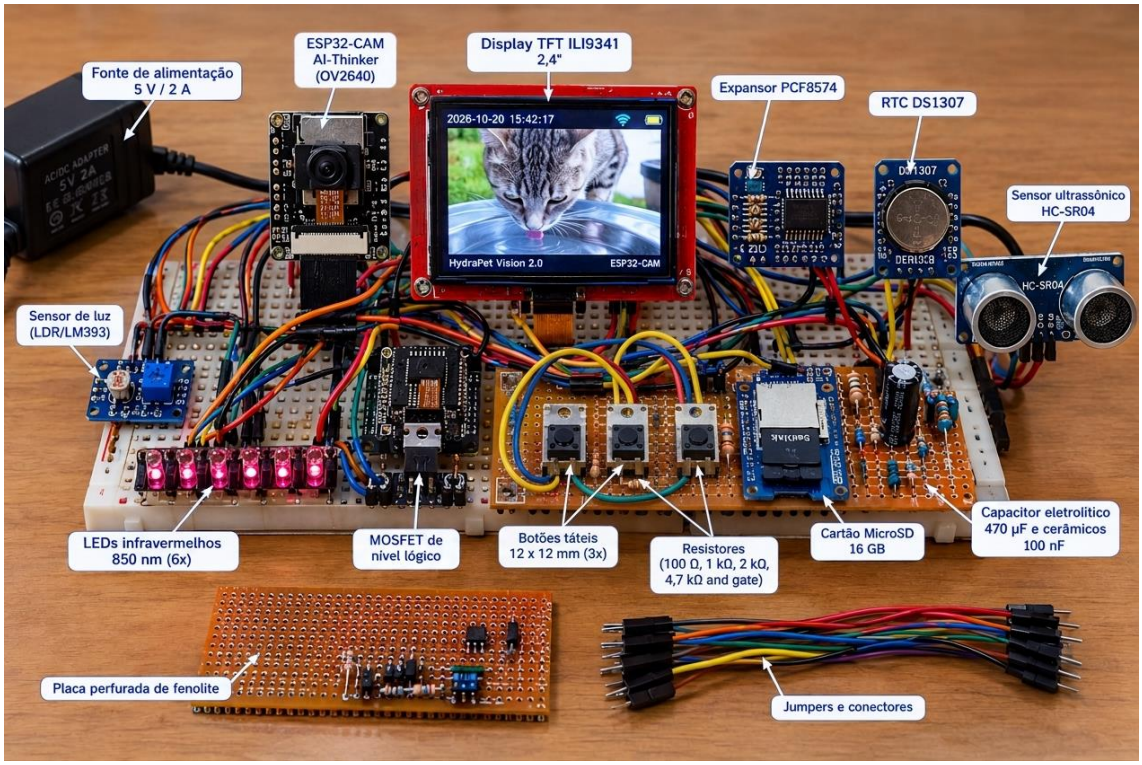
Adicionalmente, é necessário desconectar os condutores associados às GPIOs 1 e 3 durante a gravação do programa, visto que esses pinos são compartilhados com a linha DC do display TFT e com o sinal ECHO do HC-SR04. A presença de sinais externos nessas linhas durante o processo de carga interfere diretamente na comunicação UART do programador. Outro ponto crítico refere-se à GPIO 12 (MOSI do display), que deve ser mantida em nível lógico adequado durante a inicialização para evitar travamentos.

A fonte de alimentação do sistema deve ser dimensionada para suprir os picos de corrente da ESP32-CAM, do display, do leitor de cartão, do sensor ultrassônico e da iluminação infravermelha. A matriz com seis LEDs IR deve possuir resistores de limitação de corrente adequados, e o MOSFET acionador precisa ser especificado considerando a carga total dos ramos. Esses cuidados de montagem física buscam assegurar que os níveis elétricos fiquem dentro das margens operacionais exigidas pelo firmware.

3.15.1 Montagem experimental em protoboard

Antes da confecção de placas definitivas, é usual interligar os módulos de um sistema embarcado sobre uma protoboard, o que permite validar o circuito e o firmware de maneira incremental e corrigir ligações sem soldagem. A Figura 43 representa a montagem experimental prevista para o HydraPet Vision 2.0, na qual se identificam a ESP32-CAM AI-Thinker com o sensor OV2640, o display TFT ILI9341 de 2,4 polegadas, o expansor PCF8574, o módulo RTC DS1307 com sua bateria, o sensor ultrassônico HC-SR04, o módulo de luminosidade com LDR e comparador LM393, a matriz de seis LEDs infravermelhos de 850 nm acionada por MOSFET de nível lógico, os três botões táteis, o cartão MicroSD, o capacitor eletrolítico de 470 µF com capacitores cerâmicos de 100 nF, os resistores de polarização e de limitação de corrente, os jumpers de interligação e a fonte de alimentação de 5 V e 2 A. A placa perfurada de fenolite exibida à frente da protoboard destina-se à montagem do circuito auxiliar, conforme a lista de materiais apresentada na Tabela 16.

Figura 43 – Montagem experimental do HydraPet Vision 2.0 sobre protoboard, com identificação dos módulos.



Fonte: Elaborado pelos autores (2026), com auxílio de ferramenta de inteligência artificial generativa.

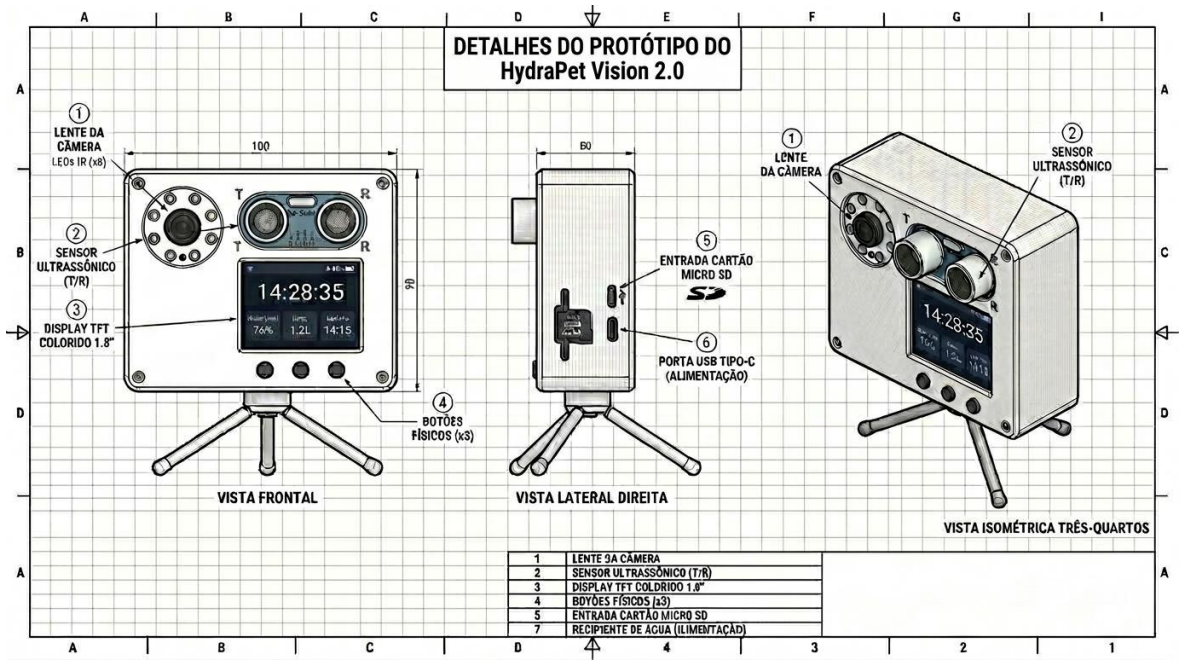
Duas observações são necessárias à leitura da figura. Primeiro, na implementação descrita neste capítulo, o cartão MicroSD é acessado pelo leitor integrado à própria ESP32-CAM, por meio do barramento SDMMC em modo de 1 bit (subseção 3.2.9); o módulo de cartão representado na ilustração deve, portanto, ser entendido apenas como indicação do meio de armazenamento. Segundo, o conteúdo exibido no display da ilustração é meramente ilustrativo, pois as telas efetivamente previstas no firmware são as descritas na seção 3.8. A figura foi produzida com ferramenta de inteligência artificial generativa a partir da especificação do circuito e tem caráter conceitual: não constitui registro fotográfico de montagem física nem evidência de validação experimental, conforme delimitado na subseção 4.1.1.

3.15.2 Invólucro e vistas de detalhamento

Para acomodar os módulos e fixá-los em relação ao recipiente de água, o projeto prevê um invólucro compacto fabricado por manufatura aditiva, pelo processo de impressão 3D FDM, conforme estabelecido nos objetivos específicos. A Figura 44 apresenta o detalhamento do conjunto em padrão de desenho técnico industrial, com vista frontal, vista lateral direita e vista

isométrica em três-quartos, acompanhadas de legenda numerada dos elementos. Na vista frontal, a lente da câmera, circundada pelos LEDs infravermelhos, e o sensor ultrassônico, com seus transdutores de emissão (T) e de recepção (R), ocupam a parte superior; o display TFT ocupa a região central; e os três botões físicos de navegação ficam alinhados na base. A vista lateral direita evidencia a entrada do cartão MicroSD e a porta USB tipo C de alimentação, posicionadas de modo que as fotografias possam ser retiradas e o equipamento energizado sem a abertura do invólucro, enquanto a vista isométrica mostra a disposição tridimensional do conjunto sobre o tripé de apoio. As cotas indicam dimensões externas da ordem de 100 mm de largura e 60 mm de profundidade.

Figura 44 – Detalhamento do protótipo em padrão de desenho técnico industrial: vistas frontal, lateral direita e isométrica.



Fonte: Elaborado pelos autores (2026), com auxílio de ferramenta de inteligência artificial generativa.

A figura é uma ilustração conceitual, produzida com ferramenta de inteligência artificial generativa, e tem caráter indicativo quanto a rótulos, cotas e proporções, não substituindo o projeto mecânico detalhado. Em particular, o display especificado e adotado no firmware é o TFT ILI9341 de 2,4 polegadas (subseção 3.2.2), prevalecendo essa especificação sobre a indicação de 1,8 polegada constante na legenda da figura, e o circuito de iluminação é composto por seis LEDs infravermelhos (subseção 3.2.7), ainda que a ilustração represente uma quantidade maior. O conteúdo exibido no display também é ilustrativo, sendo as telas previstas as descritas na seção 3.8.

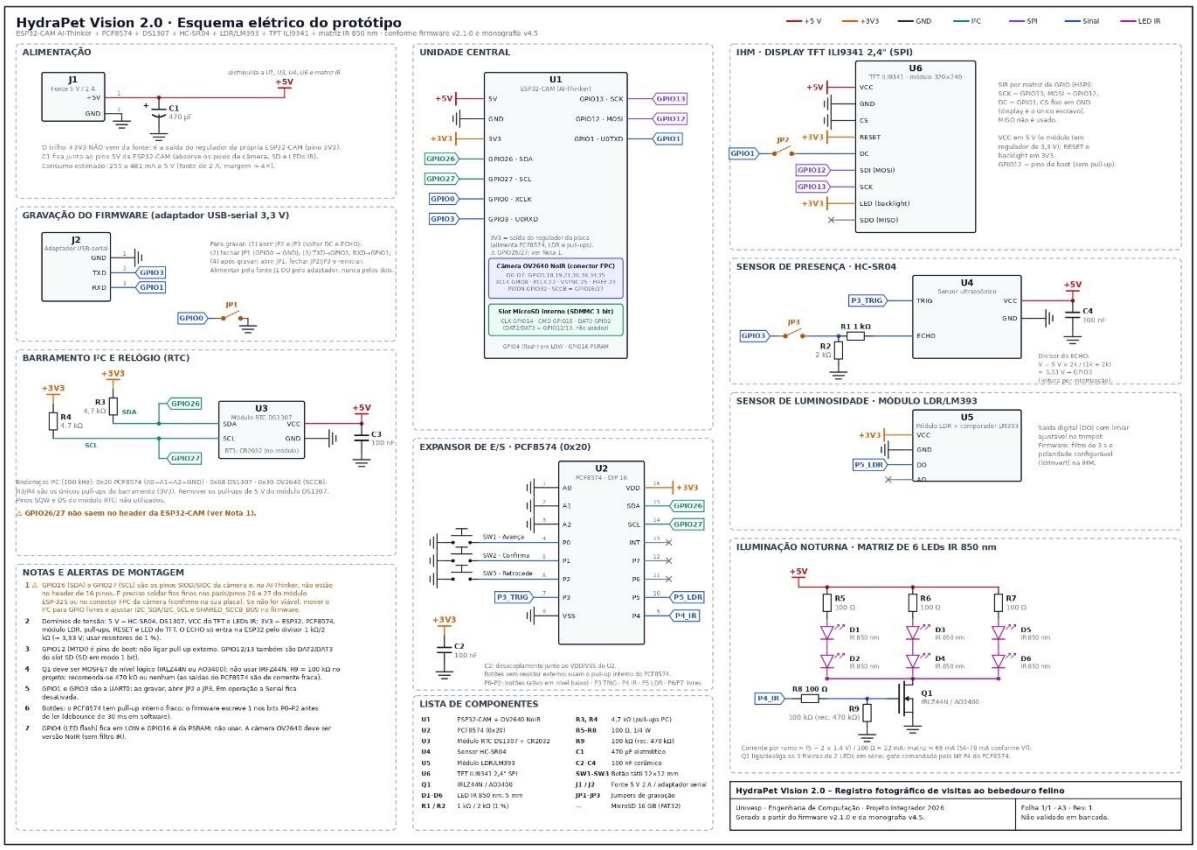
3.16 Diagrama Elétrico e Correspondência com o Firmware

O diagrama elétrico da versão 2.0 deve representar a interconexão física e a correspondência funcional com os módulos do software. O módulo ESP32-CAM posiciona-se como a unidade central de processamento. O display TFT conecta-se pelos pinos SPI: GPIO 13 (SCK), GPIO 12 (MOSI) e GPIO 1 (DC), tendo o pino CS aterrado e os pinos RST e LED ligados em 3,3 V. O cartão MicroSD interno utiliza a interface SDMMC em modo 1-bit, o que favorece a transferência de dados das fotos.

O barramento I2C compartilha a GPIO 26 (SDA) e a GPIO 27 (SCL) entre o relógio de tempo real DS1307 no endereço 0x68 e o expansor PCF8574 no endereço 0x20. O PCF8574 distribui seis sinais operacionais: P0 para o botão Avança, P1 para o botão Confirma, P2 para o botão Retrocede, P3 para o gatilho TRIG do HC-SR04, P4 para o gate do MOSFET do circuito infravermelho e P5 para a leitura do LDR. O pino ECHO do sensor ultrassônico é conectado à GPIO 3 através de um divisor de tensão.

O circuito de iluminação noturna é composto por seis LEDs IR de 850 nm organizados em três ramos paralelos, chaveados por um MOSFET acionado pelo PCF8574. Essa organização permite relacionar o esquema elétrico às funções do código fonte, auxiliando na análise do hardware. O diagrama fornece a representação completa de todas as conexões necessárias para a reprodução física do projeto e a correta validação das suas rotinas de controle e de sensoriamento.

Figura 45 – Diagrama elétrico completo do protótipo HydraPet Vision v2.0



Fonte: Elaborado pelos autores (2026).

3.17 Síntese da Evolução Entre as Versões

A versão 2.0 representa um ganho estrutural substancial em comparação com a primeira versão do protótipo. Enquanto o modelo inicial focava na validação do conceito de detecção, captura e gravação de imagens, a versão atual expandiu o projeto para uma plataforma completa de monitoramento local. O sistema passou a contar com gerenciamento de parâmetros em memória não volátil, iluminação infravermelha automática, banco de dados tabular e uma interface gráfica com galeria de imagens no display.

As principais alterações implementadas no sistema estão sintetizadas na Tabela 5, destacando a evolução física e lógica do projeto. A introdução do PCF8574 solucionou a limitação de pinos de E/S, enquanto as interrupções no sinal ECHO otimizaram o tempo de resposta do sistema. O filtro de leituras consecutivas reduziu o disparo por falsos positivos, e a inclusão dos LEDs IR e do LDR viabilizou o funcionamento do protótipo em ambientes escuros, aumentando a confiabilidade dos registros.

Tabela 5 – Evolução funcional entre a versão inicial e a versão 2.0.

Característica	Versão anterior	Versão 2.0
Expansão de GPIO	Limitada	PCF8574 via I2C
Botões	Estrutura simplificada	Três botões
Navegação	Básica	Menu estruturado
Sensor ultrassônico	Leitura bloqueante	ECHO por interrupção
Confirmação de presença	Leitura simples	Leituras consecutivas configuráveis
Iluminação noturna	Estrutura inicial	Seis LEDs IR de 850 nm
Controle da iluminação	Manual/limitado	Automático, ligado ou desligado
Sensor de luminosidade	Não estruturado	LDR + LM393
Configurações	Fixas no firmware	Persistentes em NVS
Intervalo entre fotos	Valor fixo	Configurável
Distância de detecção	Valor fixo	Configurável
Filtro de presença	Limitado	Configurável
Armazenamento	Fotografias	Fotografias + CSV
Organização das fotos	Arquivo individual	Diretório /fotos
Histórico	Limitado	Banco local navegável
Galeria	Não estruturada	Visualização no TFT
Data e hora	RTC	RTC + ajuste pela IHM
Diagnóstico	Básico	Tela de informações
Recuperação do SD	Limitada	Tentativa automática de remontagem
Interface	Menu simples	IHM estruturada em múltiplas telas

Fonte: Elaborado pelos autores (2026).

A comparação dos dados demonstra que o avanço da versão 2.0 ocorreu em nível de hardware e de software. O barramento I2C expandiu as capacidades da placa, o registro em arquivo CSV adicionou organização aos metadados e a memória NVS tornou os parâmetros permanentes. A nova interface homem-máquina transformou o display em um centro de controle e diagnóstico, permitindo ao usuário ajustar parâmetros e visualizar fotos diretamente no dispositivo sem a necessidade de leitores externos.

3.18 Síntese do Funcionamento Técnico

A partir da arquitetura apresentada, conclui-se que o funcionamento do HydraPet Vision 2.0 fundamenta-se em um ciclo contínuo de aquisição, processamento, decisão e gravação. A aquisição inicia-se com o sensor ultrassônico, que mede a distância até o animal presente na área de monitoramento. O firmware calcula o tempo de propagação da onda e verifica se o

objeto está na zona de interesse, aplicando o filtro de leituras consecutivas para evitar o acionamento indevido por ruídos.

Em paralelo, o módulo LDR monitora a luminosidade do ambiente e fornece a informação para que o software decida sobre o uso da iluminação infravermelha. Confirmada a presença, o sistema ativa os LEDs IR quando necessário e dispara a captura na câmera OV2640. O quadro JPEG é transferido para a memória PSRAM e gravado no cartão MicroSD, enquanto o RTC fornece o registro de data e hora para salvar a imagem e incluir a nova linha de registro no arquivo .CSV.

A interface gráfica recebe as atualizações do sistema e permite ao usuário acompanhar o status, consultar o histórico e alterar os parâmetros de operação. A memória NVS garante a retenção das configurações e o cartão MicroSD armazena os dados dos eventos. Essa integração entre os componentes tem por objetivo um funcionamento autônomo e abre caminho para expansões futuras, como a transmissão sem fio e o processamento de imagens por inteligência artificial.

Dessa forma, o trabalho realizado apresenta o HydraPet Vision como projeto de um registrador fotográfico de visitas ao bebedouro felino. A arquitetura prevê o atendimento dos requisitos de projeto (Tabela 7), endereça os gargalos técnicos da versão inicial e oferece uma base para a coleta de dados de campo, condicionada à montagem e aos ensaios. O sistema associa integração embarcada, registro autônomo local e flexibilidade de operação; o consumo é estimado (Seção 4.5.3), e não medido, e a autonomia por baterias seria de poucas horas, de modo que não se pode falar em baixo consumo.

3.19 Revisão 2.1 do Firmware

O código-fonte que acompanha este trabalho corresponde à revisão 2.1.0 do firmware (constante FW_VERSION). Ela mantém a arquitetura, o mapeamento de pinos e as telas descritos nas seções anteriores e incorpora correções e esclarecimentos de projeto identificados em revisão crítica. As Seções 3.1 a 3.18 e o Capítulo 4 descrevem a linha de base da versão 2.0; esta seção registra o que mudou na revisão 2.1 e prevalece em caso de divergência. A revisão foi elaborada e conferida por leitura crítica do código e compilada no Arduino IDE sem erros (Figura 49, Seção 4.7); sua execução em hardware ainda não foi realizada e integra o plano de ensaios da Seção 4.10.

Visita e fotografia. A versão 2.0 tratava cada fotografia como um “evento”: um gato parado diante do sensor gerava uma nova imagem a cada intervalo de cooldown, e o contador

crescia sem relação com o comportamento do animal. Na revisão 2.1, a unidade de registro é a visita, que se inicia quando a presença é confirmada e termina após 8 s (VISITA_GAP_MS) sem leituras dentro da faixa. Durante a visita, as fotos continuam limitadas pelo intervalo mínimo (padrão de 10 s) e por um máximo de seis imagens (MAX_FOTOS_VISITA). Ao término, uma linha é gravada em /visitas.csv com identificador, data, hora de início, duração e número de fotos; o identificador é persistido na NVS (chave vis) para não se repetir após reinício. A tela inicial passa a exibir “VISITAS HOJE” e “FOTOS”, e o arquivo /registros.csv ganha a coluna visita, que liga cada foto à sua visita. Os valores de 8 s e de seis fotos são escolhas de projeto, ainda não calibradas por observação (Seção 4.13).

Organização por data. As fotografias passam a ser gravadas em /fotos/AAAA-MM-DD/, e o nome do arquivo mantém o formato gato_AAAAMMDD_HHMMSS.jpg. Assim, nenhum diretório acumula todas as imagens, o que evita a lentidão de diretórios muito grandes e o limite de entradas por diretório do sistema FAT (MICROSOFT CORPORATION, 1999; CHAN, [s.d.]), que corresponde a 65.536 entradas de 32 bytes e se reduz quando há nomes longos.

Contadores e virada do dia. A contagem de fotos e de visitas do dia é recalculada na virada da data (função dayWatch()) e após o ajuste manual do relógio, e não apenas após uma captura, como na versão 2.0. O cálculo considera os registros mantidos em RAM (até 150), suficientes para o uso diário previsto; o total de fotos é obtido pela contagem das linhas do CSV na partida.

Diagnóstico do sensor ultrassônico. A rotina de interrupção passa a contar as bordas de subida do ECHO. Como o HC-SR04 eleva o ECHO após cada disparo, mesmo sem obstáculo, a ausência de bordas em 50 ciclos consecutivos (SENSOR_FAULT_N, alguns segundos) caracteriza falha de fiação ou do módulo e faz a tela inicial exibir “ERRO: sensor HC-SR04”. Na partida, um autoteste dispara o sensor e verifica se o ECHO responde; o indicador do boot deixa de depender apenas do expansor PCF8574, como ocorria na versão 2.0.

Armazenamento e partida. O CSV deixa de ser lido por inteiro em um objeto String: a partida conta as linhas com um buffer fixo de 512 bytes e interpreta apenas o trecho final de 20 kB (LOG_TAIL_BYTES), o que limita tempo e memória independentemente do tamanho do arquivo. A ocupação do cartão é lida uma única vez na montagem, pois usedBytes() pode levar segundos, e depois é estimada a cada foto; a tela de sistema indica o valor com o prefixo “~”. A remontagem do cartão, que bloqueia o laço por centenas de milissegundos, é tentada a cada 15 s e somente fora de uma visita.

Imagem e barramento da câmera. A decodificação para o display escolhe automaticamente a escala (1, 1/2, 1/4 ou 1/8) conforme a resolução do JPEG, o que permite configurar a captura em VGA (opção CAPTURE_FRAME_SIZE) e ainda exibi-la em 320 × 240; a configuração analisada neste trabalho permanece QVGA. Se a câmera não inicializar com o SCCB compartilhado (SHARED_SCCB_BUS 1), o firmware refaz a inicialização com o barramento de controle separado (ESPRESSIF SYSTEMS, [s.d.b]).

Formato do CSV. O separador de campos continua sendo o ponto e vírgula, e a distância passa a ser gravada com vírgula decimal, de modo que o arquivo seja interpretado corretamente por planilhas configuradas para o português do Brasil. Cartões que já contenham um registros.csv da versão 2.0 devem ter o arquivo renomeado, para manter as colunas consistentes.

Observações de montagem. O cabeçalho do código passou a registrar que as GPIOs 12 e 13, usadas como MOSI e SCK do display, são também os sinais DAT2 e DAT3 do slot MicroSD. Como o cartão opera em modo de 1 bit, esses sinais não são usados pelo cartão, o que deve ser confirmado na montagem, assim como a ausência de pull-up na GPIO 12, que define a tensão da memória flash na partida. A saída de depuração pela porta Serial permanece desativada, pois a GPIO 1 é o sinal DC do display.

Pausas bloqueantes. O laço continua cooperativo. As pausas bloqueantes são o atraso de 1 ms ao final de cada iteração, o pré-aquecimento de 300 ms do infravermelho em capturas noturnas sem antecipação, o autoteste do sensor na partida (até cerca de 0,3 s) e a remontagem do cartão, fora de visitas. A análise de latência da Seção 4.5 foi feita para a versão 2.0 e permanece válida para o ciclo de captura, pois a revisão não alterou resolução, qualidade JPEG nem a sequência de captura.

4 RESULTADOS E DISCUSSÕES

Este capítulo apresenta os resultados da avaliação conceitual do protótipo *HydraPet Vision* v2.0, abrangendo a verificação da arquitetura de hardware, o comportamento temporal do firmware, o dimensionamento elétrico, o consumo estimado, a estrutura de armazenamento e o levantamento de custos. A análise foi conduzida sobre o projeto completo, isto é, o esquema elétrico e o código-fonte em C++ descritos no Capítulo 3, e busca responder se a solução proposta é tecnicamente coerente, eletricamente viável e capaz de cumprir os requisitos definidos.

Faz-se, desde já, uma delimitação essencial o dispositivo físico não foi montado. O trabalho se encerra na etapa de projeto e de simulação conceitual do funcionamento, de modo que os números aqui apresentados não são medições realizadas em bancada nem ensaios com animais. Eles resultam de análise do código-fonte, de cálculos de engenharia fundamentados em folhas de dados e em leis físicas elementares, do rastreamento lógico de cenários de uso e de estimativas de projeto, sempre identificadas como tal. Essa distinção é mantida ao longo do capítulo e orienta a redação: onde há cálculo, informa-se a equação; onde há estimativa, informa-se a hipótese; e o que só pode ser confirmado com o equipamento montado é reunido em um plano de validação experimental (Seção 4.10), que serve de roteiro para a continuidade do trabalho.

O protótipo foi concebido para operar de forma autônoma e *offline*, o que o torna independente da rede local e mantém as imagens do ambiente doméstico sob controle do tutor. Como o SoC ESP32 dispõe nativamente de rádios Wi-Fi 802.11 b/g/n e Bluetooth, a arquitetura e o formato dos dados foram organizados de maneira modular, permitindo que a conectividade seja incorporada em versões futuras sem alteração do esquema elétrico (Seção 4.11).

4.1 Estratégia de Verificação

Esta seção define a metodologia de validação adotada para o protótipo, estruturando a avaliação em níveis progressivos, desde a análise estática do código e cálculos de engenharia até a simulação conceitual, além de estabelecer os critérios de coerência funcional, viabilidade elétrica, temporal, de recursos e econômica.

O trabalho abrange a análise estática do código-fonte, a verificação analítica por meio de cálculos e a simulação conceitual de cenários, deixando os ensaios físicos em bancada como uma etapa futura descrita na Tabela 17 (Seção 4.10). Toda essa análise fundamenta-se na

coerência funcional, na viabilidade elétrica e temporal, na suficiência de recursos e na viabilidade econômica da solução proposta.

4.1.1 Natureza da avaliação e delimitação do alcance

Em projetos de sistemas embarcados, a verificação costuma ser realizada em níveis progressivos, do mais abstrato ao mais próximo do produto final (LEE; SESHIA, 2017). Neste trabalho, os três primeiros níveis foram executados, e o quarto, que exige o hardware montado, foi deliberadamente deixado como etapa futura. A Tabela 6 apresenta essa organização e o que cada nível permite afirmar.

Tabela 6 – Níveis de verificação adotados e alcance de cada um

Nível	Método	O que permite afirmar	Situação
1	Análise estática do código-fonte	Coerência entre requisitos, constantes, máquinas de estados e tratamento de falhas; ausência de bloqueios longos no ciclo principal	Executado
2	Verificação analítica (cálculo)	Viabilidade elétrica (níveis lógicos, correntes, potência) e temporal (largura de pulso, tempos de validação e de captura)	Executado
3	Simulação conceitual de cenários	Comportamento esperado do firmware diante de situações de uso normais e de falha, por rastreamento do fluxo lógico	Executado
4	Ensaio físico em bancada e em campo	Desempenho real: taxa de detecção, qualidade das imagens, consumo medido, estabilidade elétrica	Não realizado (Seção 4.10)

Fonte: Elaborado pelos autores (2026).

Ressalta-se que plataformas de simulação virtual de circuitos, como Tinkercad Circuits, Wokwi ou Proteus Design Suite, em geral não reproduzem o comportamento do sensor de imagem OV2640, do controlador de DMA da câmera do ESP32 nem da memória PSRAM associada à placa AI-Thinker. Por essa razão, a simulação conceitual apresentada na Seção 4.7 concentra-se na lógica do firmware, e a avaliação da captura de imagem depende, em última instância, do ensaio físico.

4.1.2 Critérios de avaliação

Os resultados foram organizados segundo cinco critérios, derivados dos requisitos do Capítulo 3: (a) coerência funcional, isto é, se cada requisito possui um mecanismo

correspondente no firmware; (b) viabilidade elétrica, ou seja, se níveis de tensão, correntes e alimentação respeitam os limites dos componentes; (c) viabilidade temporal, verificando se a sequência de detecção, captura e registro cabe em intervalos compatíveis com o comportamento do animal; (d) suficiência de recursos, que engloba armazenamento, memória e energia; e (e) viabilidade econômica, avaliada pelo custo de materiais.

4.2 Verificação da Arquitetura de Hardware

Esta seção realiza a rastreabilidade dos requisitos do projeto frente aos mecanismos implementados no firmware e analisa aspectos críticos de hardware, como o compartilhamento do barramento I2C, a compatibilidade de níveis lógicos do sensor ultrassônico e o dimensionamento do circuito de iluminação infravermelha.

A Tabela 7 mostra que cada requisito possui um mecanismo correspondente no código ou foi encaminhado para verificação física (Tabela 17). O projeto analisa o compartilhamento das linhas SDA e SCL entre a câmera OV2640, o expansor PCF8574 e o relógio DS1307 operando em endereços distintos, além de avaliar o divisor resistivo para adequar o sinal de 5V do HC-SR04 aos limites de 3,3V do ESP32, dimensionando também as correntes e os componentes de chaveamento da matriz de LEDs infravermelhos.

4.2.1 Rastreabilidade dos requisitos

Os requisitos do projeto, derivados dos objetivos específicos (Seção 1.2.2) e do levantamento com tutores (Seção 2.4.1), estão definidos no Quadro 1. A coluna de critério indica como cada um é considerado satisfeito; a rastreabilidade para o firmware consta da Tabela 7.

Quadro 1 – Requisitos funcionais (RF) e não funcionais (RNF) do projeto

Requisito	Descrição	Critério de aceitação
RF01	Detectar a aproximação do animal ao recipiente por sensor ultrassônico.	Presença confirmada após N leituras consecutivas na faixa (padrão: 4 leituras, até 15 cm).
RF02	Registrar fotografia durante a visita.	Arquivo JPEG gravado no cartão SD com nome único; no máximo seis fotos por visita.

RF03	Associar cada registro a data e hora do relógio de tempo real.	Data e hora presentes no nome do arquivo e no CSV.
RF04	Operar com pouca luz, com iluminação infravermelha de 850 nm acionada automaticamente.	LEDs acesos após 3 s de escuro estável e imagem utilizável (a ensaiar).
RF05	Exibir o estado do sistema na tela inicial.	Relógio, visitas, fotos, distância e indicadores de SD, IR, luminosidade e falhas.
RF06	Permitir a consulta de registros e fotos no próprio display.	Lista de registros e visualizador acessíveis pelos três botões.
RF07	Permitir o ajuste de data, hora e parâmetros, mantidos sem energia.	Valores preservados após reinício (RTC e NVS).
RF08	Gravar os dados em formato aberto, legível em planilhas.	registros.csv e visitas.csv com cabeçalho e separador ponto e vírgula.
RNF01	Operar offline, sem enviar imagens à rede.	Nenhum uso de Wi-Fi ou Bluetooth no firmware.
RNF02	Manter o sistema responsivo, sem bloqueios longos no ciclo principal.	Pausas bloqueantes pontuais e documentadas (Seção 4.5).
RNF03	Ter custo reduzido, com componentes comerciais.	Custo de materiais estimado (Seção 4.9).
RNF04	Não perturbar o animal.	Iluminação infravermelha; percepção do ultrassom a avaliar (Seção 4.13).
RNF05	Tolerar falhas de periféricos e sinalizá-las ao usuário.	Estados de erro de câmera, SD, expansor e sensor exibidos na tela inicial.

Fonte: Elaborado pelos autores (2026).

A Tabela 7 relaciona cada requisito do projeto ao mecanismo do firmware que o implementa e ao método pelo qual foi verificado. Todos os requisitos funcionais possuem implementação identificável no código; aqueles que dependem do desempenho real dos componentes são explicitamente marcados como pendentes de ensaio físico.

Tabela 7 – Rastreabilidade entre requisitos, firmware e verificação

Requisito	Mecanismo no firmware	Verificação	Situação
RF01	Interrupção no ECHO, máquina de estados do HC-SR04 e limiar ajustável (processDistance)	Análise e cálculo	Previsto e verificado analiticamente
RF02	doCapture(): quadro JPEG gravado em /fotos/AAAA-MM-DD	Análise e simulação	Previsto e verificado analiticamente
RF03	getNow() lê o RTC; nome do arquivo e campos do CSV usam o registro temporal	Análise e simulação	Previsto e verificado analiticamente
RF04	lightUpdate() e irTask(): LEDs IR por LDR filtrado por 3 s	Análise e simulação	Qualidade da imagem pendente de ensaio
RF05	Tela inicial com relógio, contadores de visitas e fotos, distância e indicadores	Análise	Previsto e verificado analiticamente
RF06	Menu de 5 itens, lista de registros e visualizador de fotos (TJpg_Decoder)	Análise e simulação	Tempo de exibição pendente de ensaio
RF07	Edição de data e hora (rtc.adjust) e parâmetros na NVS (Preferences)	Análise e simulação	Previsto e verificado analiticamente
RF08	appendCsv() e endVisit(): registros.csv (por foto) e visitas.csv (por visita), delimitados por ponto e vírgula	Análise	Previsto e verificado analiticamente
RNF01	Nenhum uso de rede; armazenamento local em MicroSD e NVS	Análise	Previsto e verificado analiticamente
RNF02	Ciclo cooperativo; ECHO por interrupção; pausas bloqueantes pontuais (1 ms, 300 ms de pré-aquecimento IR e remontagem do SD fora de visitas)	Análise (Seção 4.5)	Previsto e verificado analiticamente
RNF03	Componentes comerciais e IDE Arduino (Seção 4.9)	Levantamento de custos	Previsto e verificado analiticamente
RNF04	LEDs de 850 nm; o HC-SR04 emite ultrassom de 40 kHz, possivelmente audível a felinos (Seção 4.13)	Análise	Percepção do animal pendente de observação
RNF05	Variáveis camOk, sdOk, rtcOk, pcfOk e sensorFault; novas tentativas do SD e do expansor; alarme do sensor	Análise e simulação	Previsto e verificado analiticamente

Fonte: Elaborado pelos autores (2026).

4.2.2 Alocação de pinos e barramento I2C compartilhado

A escassez de GPIOs, principal limitação da versão 1.0, foi resolvida pela combinação de três recursos: o cartão MicroSD no modo SDMMC de 1 bit, o barramento I2C compartilhado com o canal de controle da câmera (SCCB) e o expansor PCF8574, cuja distribuição de portas foi apresentada no Capítulo 3. Do ponto de vista da análise de endereços, os três dispositivos que compartilham as linhas SDA (GPIO 26) e SCL (GPIO 27) respondem em endereços distintos: o sensor OV2640 em 0x30, o expansor em 0x20 e o relógio em 0x68. Não há, portanto, conflito de endereçamento, e a taxa de 100 kHz atende ao limite do DS1307.

O ponto que exige atenção é de natureza de *software*: a diretiva SHARED_SCCB_BUS, definida como 1, instrui o *driver* da câmera a reutilizar o barramento Wire já iniciado, o que depende da versão do pacote de placas ESP32. Trata-se de um risco de integração que só pode ser descartado com o equipamento montado; por isso, o código prevê a alternativa SHARED_SCCB_BUS 0 e, na revisão 2.1, refaz automaticamente a inicialização com SCCB separado caso a câmera não responda; o ensaio de reinicializações sucessivas consta do plano de validação (Quadro de ensaios da Seção 4.10).

4.2.3 Compatibilidade de níveis lógicos do sinal ECHO

O pino *Echo* do HC-SR04 opera em 5V, ao passo que os GPIOs do ESP32 admitem tensão de entrada de nível alto a partir de 0,75 × VDD, isto é, cerca de 2,5 V, e tensão máxima de VDD + 0,3 V (ESPRESSIF SYSTEMS, [s.d.a]). O divisor resistivo de 1 kΩ (R1) e 2 kΩ (R2) resulta na tensão dada pela Equação (1).

$$V_{saída} = V_{entrada} \times R2 / (R1 + R2) = 5 \times 2 / (1 + 2) \approx 3{,}33\ V \quad (1)$$

A margem para o limiar de nível alto é de aproximadamente 0,8 V. Considerando resistores com tolerância de ±5%, a tensão de saída varia entre 3,22 V e 3,44 V. Com a fonte de 5 V no limite superior de sua tolerância (5,25 V), o valor nominal sobe para 3,5 V e, combinando esse limite com as tolerâncias no sentido desfavorável, chega a cerca de 3,61 V, ligeiramente acima do máximo de 3,6 V (VDD + 0,3 V). O caso extremo é improvável, mas não é admissível em projeto: recomenda-se utilizar resistores de 1% de tolerância (pior caso de aproximadamente 3,52 V) e uma fonte bem regulada. A corrente consumida pelo divisor, de 5 V / 3 kΩ ≈ 1,7 mA, é desprezível.

4.2.4 Dimensionamento do circuito de iluminação infravermelha

Cada ramo da matriz de iluminação, com dois LEDs em série e resistor de 100 Ω, drena a corrente dada pela Equação (2), em que a queda direta Vf típica de um LED infravermelho de 850 nm é de 1,4 V, mas varia entre fabricantes.

$$I = (Vcc - 2 \times Vf) / R = (5 - 2 \times 1{,}4) / 100 = 22\ mA \quad (2)$$

Para Vf de 1,2 V e 1,6 V, a corrente resulta em 26 mA e 18 mA, respectivamente, o que leva a corrente total da matriz (três ramos) a variar de 54 mA a 78 mA. Esses valores são compatíveis com a operação contínua dos LEDs de 5 mm, cujo limite usual é superior a 50 mA por dispositivo, conforme a folha de dados do componente adquirido. A potência dissipada em cada resistor é de cerca de 48 mW, ou seja, menos de 20% da capacidade de um resistor de 1/4 W.

O acionamento por MOSFET de canal N exige que o transistor seja de **nível lógico**, isto é, plenamente saturado com tensão de *gate* de 3,3 V. Componentes como o IRLZ44N ou o AO3400 atendem a essa condição, enquanto o IRFZ44N, de uso muito difundido, possui tensão de limiar entre 2 V e 4 V e não deve ser empregado nessa configuração. Além disso, as portas do PCF8574 fornecem nível alto por meio de uma fonte de corrente fraca, da ordem de dezenas a poucas centenas de microampères (NXP SEMICONDUCTORS, 2013). Com o resistor de descarga de 100 kΩ entre *gate* e terra, previsto no esquema, a corrente necessária para manter 3,3 V no *gate* é de 33 µA, valor próximo ao mínimo garantido pelo fabricante. Recomenda-se, portanto, elevar esse resistor para valores da ordem de 470 kΩ, ou mesmo suprimi-lo, verificando o comportamento na montagem (Seção 4.8).

4.3 Sensoriamento de Proximidade e Critério de Confirmação

Esta seção detalha o funcionamento do sensor ultrassônico HC-SR04, abordando a conversão matemática entre a largura do pulso e a distância, o impacto desprezível da variação térmica na velocidade do som e a adoção de um filtro de confirmação por leituras consecutivas para evitar disparos falsos.

A Equação (3) traduz o tempo de eco em distância com resolução centimétrica, detalhada na Tabela 8 para diferentes faixas de operação. Além disso, demonstra-se que as variações térmicas típicas de ambientes domésticos causam erros pequenos na medição, definindo-se o uso padrão de quatro leituras consecutivas para conferir tolerância a ruídos conforme apresentado na Tabela 9.

4.3.1 Relação entre distância e largura do pulso

A rotina de interrupção echoISR() registra o instante das bordas de subida e de descida do sinal *Echo* com resolução de 1 µs, fornecida por micros(). A distância é obtida pela Equação (3), em que *t* é a largura do pulso em microssegundos. A Tabela 8 apresenta a largura de pulso esperada para distâncias representativas, obtida por essa relação.

$$d = t \times 0{,}0343 / 2 \qquad \text{ou} \qquad t = 2d / 0{,}0343 \tag{3}$$

Tabela 8 – Largura de pulso do sinal ECHO esperada para distâncias representativas

Distância (cm)	Largura do pulso (µs)	Observação
2	117	Limite inferior de validade (MIN_VALID_CM)
5	292	Limite mínimo do limiar configurável
10	583	—
15	875	Limiar padrão de disparo
30	1.749	Distância de antecipação do IR (2 × limiar padrão)
60	3.499	Limite máximo do limiar configurável
400	23.324	Limite de descarte de leituras (ruído)

Fonte: Elaborado pelos autores (2026), a partir da Equação (3).

O tempo limite de 35 ms adotado na espera do eco corresponde a uma distância de cerca de 6 m, superior ao alcance nominal do sensor (4 m), o que assegura que qualquer eco válido seja processado antes do descarte. Como a medição é concluída por interrupção, ela independe do momento em que o ciclo principal verifica o resultado, característica que preserva a precisão da leitura mesmo quando o processador está ocupado com o redesenho da tela.

4.3.2 Influência da temperatura

A velocidade do som no ar depende da temperatura e pode ser aproximada por $v \approx 331{,}3 + 0{,}606 \cdot T$, em m/s, com T em graus Celsius. O firmware adota o valor fixo de 343 m/s, correspondente a cerca de 20 °C. Em um ambiente doméstico de 5 °C a 35 °C, a velocidade real varia de 334,3 m/s a 352,5 m/s, ou seja, de −2,5% a +2,8% em relação ao valor adotado. Para um objeto a 15 cm, isso equivale a um erro entre −0,4 cm e +0,4 cm, desprezível diante da faixa de detecção de 2 cm a 15 cm e da própria resolução de 0,3 cm especificada para o HC-

SR04 (ELECFREAKS, [s.d.]). Não se justifica, portanto, a inclusão de sensor de temperatura para compensação.

4.3.3 Tempo de validação e escolha do número de leituras

Para impedir que reflexões espúrias, correntes de ar ou a movimentação da superfície da água disparem a câmera, o firmware exige que *N* leituras consecutivas caiam dentro da faixa válida (parâmetro confirmReads, ajustável de 2 a 10). Como as medições ocorrem a cada 60 ms, a confirmação da presença consome, após a primeira leitura válida, um tempo mínimo de $(N - 1) \times 60$ ms, apresentado na Tabela 9

Tabela 9 – Tempo mínimo de validação em função do número de leituras de confirmação

Leituras (N)	Tempo mínimo de validação (ms)	Interpretação de projeto
2	60	Filtro mínimo; maior sensibilidade a leituras espúrias isoladas
3	120	Filtro leve
4 (padrão)	180	Compromisso adotado entre rejeição de ruído e rapidez de resposta
6	300	Filtro mais restritivo; pode perder aproximações muito rápidas
8	420	Filtro conservador; tempo comparável a poucos movimentos de lambida
10	540	Limite máximo do parâmetro

Fonte: Elaborado pelos autores (2026), com período de amostragem de 60 ms (SENSOR_PERIOD_MS).

A escolha do valor padrão N = 4 baseia-se em um argumento de projeto, e não em uma medição: o tempo de validação de 180 ms é pequeno diante da duração típica de um episódio de ingestão, no qual o gato realiza várias lambidas, em frequência de cerca de 3,5 lambidas por segundo (REIS et al., 2010), e, ao mesmo tempo, exige que a presença se mantenha por quatro amostras seguidas, o que descarta ecos isolados. Como qualquer leitura fora da faixa zera o contador, o filtro é intrinsecamente conservador. A taxa efetiva de falsos disparos e de perdas para cada valor de N depende do posicionamento do sensor em relação ao recipiente e do comportamento do animal, e será determinada no ensaio físico. Para isso, o parâmetro foi disponibilizado no menu de configurações, dispensando nova compilação do código.

4.4 Iluminação Infravermelha e Qualidade de Imagem Esperada

Esta seção explora a lógica de acionamento automático e sob demanda da iluminação infravermelha com base na leitura do sensor LDR, considerando os tempos de pré-aquecimento necessários para a convergência do sensor de imagem e as características ópticas das câmeras NoIR.

Detalham-se os modos de operação da luz infravermelha listados na Tabela 10, o que inclui a retenção de 2,5 segundos e o pré-aquecimento de 300 ms voltados à estabilização do ganho e da exposição da câmera OV2640. Destaca-se também a necessidade de empregar o módulo de câmera sem filtro IR e a consequente natureza monocromática das imagens obtidas sob o comprimento de onda de 850 nm.

4.4.1 Estratégia de acionamento sob demanda

A iluminação infravermelha é controlada por duas informações: a classificação do ambiente como claro ou escuro, feita pelo módulo LDR, e a proximidade do animal. Uma mudança de estado do LDR só é aceita após três segundos de estabilidade (LDR_STABLE_MS), o que impede que sombras passageiras ou o acendimento momentâneo de uma lâmpada provoquem o acionamento intermitente dos LEDs. No modo automático, quando o ambiente está escuro e o objeto se aproxima a até o dobro do limiar (30 cm no padrão), os LEDs são mantidos acesos por 2,5 s (irHoldUntil). Se, no momento da captura, o infravermelho ainda estiver apagado, o firmware o liga e aguarda 300 ms (IR_PREHEAT_MS) antes de obter o quadro.

A justificativa para esse intervalo está no funcionamento do sensor OV2640, cujos algoritmos de controle automático de exposição (AEC) e de ganho (AGC) convergem ao longo de alguns quadros. Admitindo-se uma taxa de 20 a 30 quadros por segundo em resolução QVGA, 300 ms correspondem a seis a nove quadros, o que é, em princípio, suficiente para que a exposição se ajuste à nova iluminação. O firmware descarta, ainda, o primeiro quadro obtido, pois, com dois *framebuffers* em modo CAMERA_GRAB_LATEST, ele pode ter sido gerado antes do acendimento dos LEDs. Trata-se de um raciocínio de projeto, cuja eficácia depende do ensaio com a câmera. A Tabela 10 resume as consequências esperadas de cada modo de operação.

Tabela 10 – Modos de iluminação infravermelha e comportamento esperado

Modo (cfg.irMode)	Acionamento dos LEDs	Efeito esperado na imagem	Consumo
IR_OFF	Nunca	Em ambiente escuro, imagens sem informação útil	Menor
IR_AUTO com antecipação	Escuro e objeto a até 2 × limiar; retenção de 2,5 s	Exposição já ajustada no disparo; melhor qualidade esperada	Baixo (sob demanda)
IR_AUTO sem antecipação	Escuro e aproximação muito rápida: acende no disparo e aguarda 300 ms	Exposição ajustada durante a espera; latência maior	Baixo (sob demanda)
IR_ON	Permanente	Iluminação constante, útil para testes e ajuste do foco	Maior (+66 mA contínuos)

Fonte: Elaborado pelos autores (2026). Comportamento esperado por projeto, sujeito a confirmação em bancada.

4.4.2 Condicionantes ópticos

Três condicionantes ópticos devem ser considerados na análise. O primeiro é o filtro de corte de infravermelho (*IR-cut*) presente na lente da maioria das câmeras OV2640, que atenua a sensibilidade a 850 nm; para o funcionamento do sistema noturno, é necessário utilizar a versão sem esse filtro, disponível comercialmente como módulo *NoIR* ou de visão noturna, opção considerada no levantamento de custos. O segundo é a distância de trabalho: como o animal é fotografado a poucos centímetros do sensor, a seis LEDs concentrados próximos à câmera tendem a proporcionar irradiância suficiente, favorecida pelo alcance curto. O terceiro é a natureza monocromática da imagem: sob luz de 850 nm, o sensor produz imagens sem informação de cor, o que preserva a identificação da postura e do porte, mas não do padrão de pelagem colorido. Nenhuma dessas afirmações substitui a avaliação da nitidez e do ruído em ambiente com iluminância inferior a 1 lux, prevista no plano de validação.

4.5 Desempenho Temporal e Consumo de Energia

Esta seção analisa o comportamento temporal do sistema baseado em um ciclo de execução cooperativo e estima a latência total desde a detecção da presença animal até a gravação e exibição do registro, acompanhadas pelo orçamento detalhado de corrente e consumo elétrico.

Conforme ilustrado na Tabela 11 e na Figura 46, o ciclo completo de captura e armazenamento ocorre em menos de um segundo, variando de 0,3 a 0,7 segundos. Já a Tabela 12 e a Figura 47 quantificam o consumo nos diferentes estados operacionais do sistema, oscilando entre 255 mA e 481 mA em 5V, o que indica, por estimativa, a adequação da fonte de 2 A especificada e fundamenta a avaliação dos limites para uma eventual operação autônoma por baterias.

4.5.1 Ciclo Cooperativo e Responsividade

O firmware não utiliza sistema operacional de tempo real: as seis tarefas do ciclo principal são executadas em sequência e cada uma retorna imediatamente caso não haja trabalho a fazer. As pausas bloqueantes são o atraso de 1 ms ao final da iteração, o pré-aquecimento de 300 ms, restrito a capturas noturnas sem antecipação, o autoteste do sensor na partida e a remontagem do cartão SD, esta última permitida apenas fora de visitas. O trecho mais longo do ciclo é o redesenho integral de uma tela, cuja transferência de 153.600 bytes a 26 MHz leva, no mínimo, cerca de 47 ms. Durante esse período, o eco do sensor continua sendo capturado pela interrupção, e o resultado é processado na iteração seguinte, sem perda da medição. O efeito máximo é o atraso de uma amostragem de 60 ms, o que não compromete a validação da presença. Por essa razão, o atendimento ao requisito RNF02 dispensou a divisão do código em tarefas de FreeRTOS distribuídas entre os dois núcleos, opção que permanece disponível caso o ensaio revele latências inaceitáveis.

4.5.2 Latência do ciclo de detecção ao registro

A Tabela 11 estima a duração de cada etapa do processo, desde o início da confirmação da presença até a exibição da pré-visualização. As faixas foram construídas com base no período de amostragem do sensor (valor exato), nas taxas nominais da interface SDMMC e do barramento I2C e no comportamento típico da câmera, e devem ser tomadas como ordens de grandeza a serem confirmadas por medição.

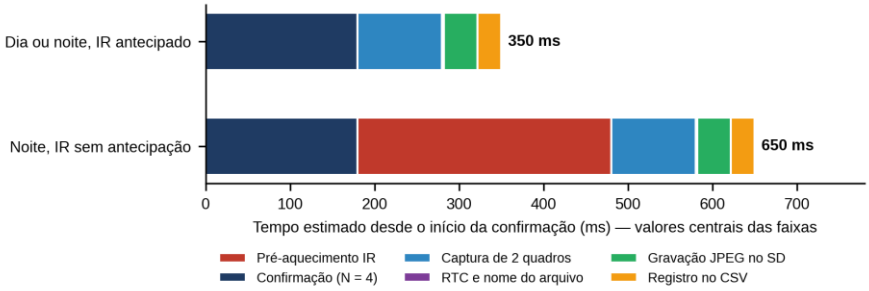
Tabela 11 – Estimativa de latência do ciclo de captura e registro

Etapa	Mínimo (ms)	Máximo (ms)	Base da estimativa
Confirmação da presença (N = 4)	180	180	(N − 1) × 60 ms; valor exato
Pré-aquecimento do IR	0	300	Zero com antecipação; 300 ms sem antecipação
Descarte e captura de quadros	70	130	Dois quadros a 20–30 quadros/s
Leitura do RTC e nome do arquivo	1	3	Leitura de poucos bytes por I2C a 100 kHz
Gravação do JPEG no SD	20	60	Arquivo de dezenas de kB por SDMMC de 1 bit
Acréscimo ao registros.csv	15	40	Abertura, escrita e fechamento do arquivo
Subtotal até o registro concluído	286	713	Soma das etapas anteriores
Decodificação e exibição da foto	100	300	Transferência mínima de 47 ms mais a decodificação
Total até a pré-visualização	386	1.013	Soma geral

Fonte: Elaborado pelos autores (2026). Estimativas de projeto, não medições.

Em qualquer situação, o registro é concluído em menos de um segundo após a primeira leitura válida, e, com o infravermelho antecipado, em cerca de 0,3 a 0,4 s. A Figura 46 ilustra a composição da latência nos dois casos noturnos extremos.

Figura 46 – Composição estimada da latência de captura e registro



Fonte: Elaborado pelos autores (2026).

4.5.3 Orçamento de consumo e alimentação

Na ausência de medições, o consumo foi estimado a partir de valores típicos de folhas de dados e de valores conservadores de projeto, agrupados por bloco funcional: 150 mA para a

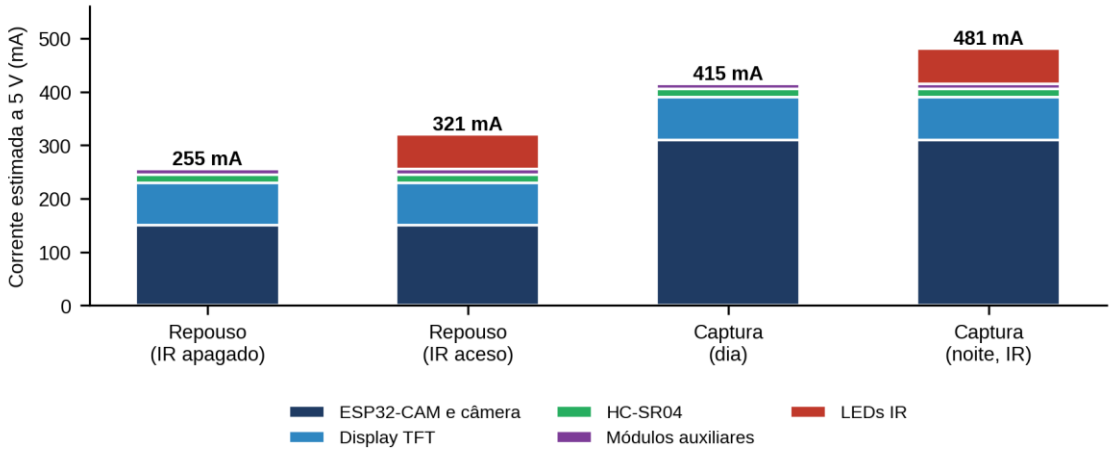
ESP32-CAM com câmera ativa e sem Wi-Fi em regime de repouso, elevados a 310 mA nos picos de captura e gravação; 80 mA para o display com luz de fundo; 15 mA para o HC-SR04; 10 mA para RTC, expansor e módulo LDR; e 66 mA para a matriz de LEDs infravermelhos (Seção 4.2.4). A Tabela 12 e a Figura 47 apresentam o resultado para quatro estados de operação.

Tabela 12 – Orçamento de consumo estimado do protótipo (alimentação de 5 V)

Estado de operação	Blocos ativos	Corrente (mA)	Potência (W)
Repouso, IR apagado	ESP32-CAM, display, HC-SR04, auxiliares	255	1,28
Repouso, IR aceso	Anterior mais matriz de LEDs IR	321	1,61
Pico de captura, dia	Câmera e SD em pico, display, HC-SR04, auxiliares	415	2,08
Pico de captura, noite	Anterior mais matriz de LEDs IR	481	2,41

Fonte: Elaborado pelos autores (2026). Valores de projeto, sujeitos a confirmação por medição.

Figura 47 – Composição estimada da corrente por estado de operação



Fonte: Elaborado pelos autores (2026).

A fonte de 5 V e 2 A prevista dispõe de margem de cerca de 4 vezes em relação ao pico de 481 mA, o que atende à recomendação de folga para evitar quedas de tensão (*brownout*), problema recorrente na plataforma, agravado pela ocorrência simultânea de picos da câmera, da gravação no cartão e da matriz de LEDs. O capacitor de 470 µF junto à placa contribui para suavizar esses picos. Em regime de repouso, a potência de 1,28 W corresponde a um consumo de aproximadamente 30,6 Wh por dia, ou 0,03 kWh.

A estimativa também relativiza a alimentação por baterias. Uma célula 18650 de 3.000 mAh (11,1 Wh), com eficiência de conversão de 90%, forneceria cerca de 7,8 h de operação no estado de repouso. A autonomia de dias, portanto, não é alcançada apenas com o acréscimo de baterias: exigiria reduzir o consumo do display, por exemplo com controle da luz de fundo e apagamento por inatividade, e explorar modos de baixo consumo do microcontrolador, medidas que constam das melhorias futuras.

4.6 Registro, Armazenamento e Banco de Dados Local

Esta seção descreve o mecanismo de persistência local de dados, detalhando a estrutura dos arquivos de imagem JPEG e do arquivo tabular em formato CSV, a capacidade de armazenamento de longa duração em cartões MicroSD e as estratégias para garantir a integridade dos registros.

Apresenta-se o formato exemplificado na Figura 48 juntamente com a rotina de cache em memória RAM dos eventos mais recentes para agilizar a navegação diretamente no display. Estima-se na Tabela 13 que cartões de 16 GB comportam anos de operação nas hipóteses adotadas, ao passo que o fluxo de gravação prioriza estritamente a consistência entre o arquivo de imagem gerado e as linhas adicionadas ao banco tabular CSV.

4.6.1 Estrutura dos dados persistidos

Cada foto gera dois registros: o arquivo de imagem em /fotos, com nome no formato gato_AAAAMMDD_HHMMSS.jpg, e uma linha no arquivo /registros.csv, com os campos data, hora, arquivo, distância e uso do infravermelho (a revisão 2.1 acrescenta a coluna visita e o arquivo /visitas.csv). A Figura 48 mostra o formato da versão 2.0 com dados hipotéticos, elaborados apenas para ilustrar a estrutura.

Figura 48 – Exemplo ilustrativo do conteúdo de registros.csv (dados hipotéticos)

```
data;hora;arquivo;distancia_cm;ir
2026-09-20;22:10:30;/fotos/gato_20260920_221030.jpg;12.4;1
2026-09-20;22:14:15;/fotos/gato_20260920_221415.jpg;10.8;1
2026-09-21;07:30:02;/fotos/gato_20260921_073002.jpg;14.1;0
```

Fonte: Elaborado pelos autores (2026).

Na inicialização, o firmware lê o CSV e mantém em memória os 150 registros mais recentes, em um vetor de estruturas LogEntry de 72 bytes cada na revisão 2.1 (60 bytes na versão 2.0), ocupando cerca de 10,8 kB de RAM. Isso permite navegar pela lista sem acessar o cartão a cada movimento do cursor. Os cinco parâmetros configuráveis são gravados na partição NVS da memória flash pela biblioteca *Preferences*, apenas quando o usuário confirma a alteração, o que limita os ciclos de escrita e preserva os ajustes após a perda de energia.

4.6.2 Capacidade de armazenamento

Imagens JPEG em resolução QVGA com qualidade 10 costumam ocupar algumas dezenas de quilobytes ou menos, mas o sistema de arquivos FAT32 aloca espaço em clusters, tipicamente de 8 KB a 32 KB conforme a formatação. Na prática, cada foto ocupa ao menos um cluster, e, de forma conservadora, adotam-se ocupações de 16 KB e de 32 KB por foto. A Tabela 13 apresenta a capacidade de um cartão de 16 GB e a duração equivalente para três taxas de eventos por dia.

Tabela 13 – Capacidade estimada de um cartão MicroSD de 16 GB

Ocupação por foto	Fotos armazenáveis	50 eventos/dia	100 eventos/dia	200 eventos/dia
16 KB	976.562	53,5 anos	26,7 anos	13,4 anos
32 KB	488.281	26,7 anos	13,4 anos	6,7 anos

Fonte: Elaborado pelos autores (2026). Considera 16×10^9 bytes e desconsidera o espaço do CSV.

A capacidade, portanto, não é o fator limitante. Os limites práticos são outros: o tempo de leitura do CSV a cada inicialização cresce linearmente com o número de linhas, e diretórios FAT32 com centenas de milhares de arquivos tornam-se lentos. Essas duas medidas, a leitura apenas do trecho final do CSV e a criação de subpastas por data (cujo limite de entradas por diretório decorre do sistema FAT; MICROSOFT CORPORATION, 1999; CHAN, [s.d.]), foram incorporadas à revisão 2.1 do firmware (Seção 3.19).

4.6.3 Integridade dos registros

A ordem das operações foi escolhida para preservar a consistência: o arquivo de imagem é gravado e tem seu tamanho conferido antes de a linha ser acrescentada ao CSV, de modo que cada registro aponta para uma imagem completa. Como o CSV é aberto, escrito e fechado a cada evento, a janela de exposição a uma interrupção de energia é curta. Não se pode, contudo,

afirmar imunidade a falhas de alimentação: um corte durante a gravação pode deixar uma imagem incompleta ou uma linha truncada, situação cuja frequência só pode ser determinada por ensaio (Seção 4.10). Caso a gravação falhe, o cartão é marcado como defeituoso, o usuário é avisado e a montagem é tentada novamente a cada 15 segundos, fora de visitas.

4.7 Simulação Conceitual de Cenários de Uso

Para avaliar a lógica do firmware, foram rastreados nove cenários representativos, percorrendo-se o código desde a condição de entrada até o estado final do sistema. A Tabela 14 apresenta o resultado esperado em cada caso. Por se tratar de rastreamento lógico, o resultado descreve o que o firmware faz por construção, e não o desempenho dos componentes físicos.

Tabela 14 – Cenários simulados e comportamento esperado do firmware

Cenário	Condição de entrada	Comportamento esperado
C1 – Aproximação diurna	Ambiente claro; quatro leituras consecutivas entre 2 cm e 15 cm; SD operante	Captura sem IR; JPEG gravado em /fotos/AAAA-MM-DD; visita aberta; linha no CSV; contadores atualizados; pré-visualização de 3 s
C2 – Passagem rápida	Apenas duas leituras dentro da faixa, seguidas de leitura fora	Contador de acertos zerado; nenhuma captura
C3 – Aproximação noturna	Escuridão estável por 3 s; objeto a até 30 cm e depois a 15 cm	IR retido por 2,5 s; captura com ir = 1; LEDs apagados cerca de 1,5 s após o disparo
C4 – Permanência prolongada	Animal imóvel dentro da faixa após a captura	Nova captura somente após o intervalo mínimo (10 s no padrão); sem fotos repetidas antes disso e com no máximo seis fotos por visita
C5 – Cartão SD ausente	Cartão removido ou com falha de montagem	Indicador “SD ERRO”; captura abandonada com aviso; nova montagem a cada 15 s (fora de visitas); registros recarregados ao restabelecer
C6 – RTC não encontrado	Módulo desconectado ou sem resposta	Aviso na tela inicial; relógio de contingência baseado em millis(); arquivos nomeados sem colisão
C7 – Expansor sem resposta	Falha de leitura I2C no PCF8574	Sensor e botões suspensos; mensagem “ERRO: expansor I/O”; nova tentativa a cada 2 s; câmera e display seguem ativos
C8 – Ajuste de data e hora	Usuário edita os campos e escolhe Salvar	rtc.adjust() com segundos zerados; contagem do dia recalculada; confirmação na tela

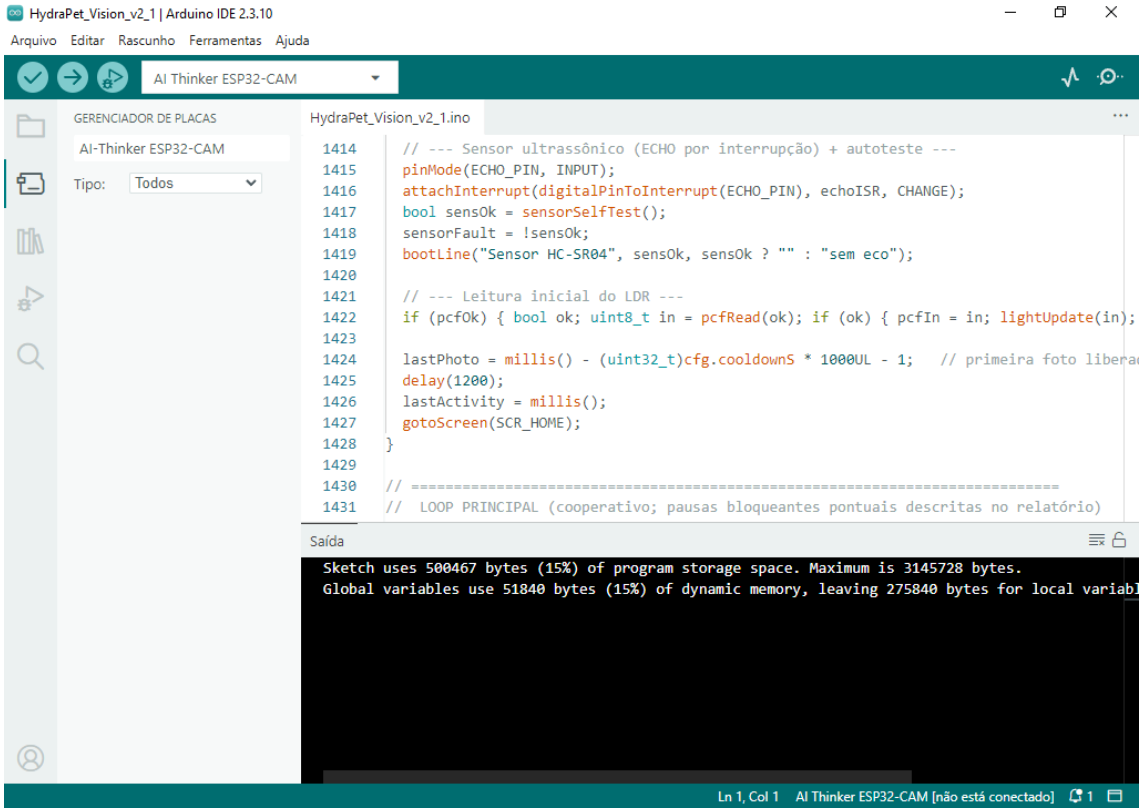
C9 – Reinicialização	Perda e retorno da alimentação	Parâmetros recuperados da NVS; registros recarregados do CSV; hora mantida pela bateria do RTC

Fonte: Elaborado pelos autores (2026).

O rastreamento mostrou que, nos nove cenários, o firmware conduz o sistema a um estado coerente com o projeto e que nenhuma falha de periférico interrompe as demais funções, o que corrobora o atendimento ao requisito RNF05. Observou-se ainda que, no cenário C5, o intervalo mínimo entre fotos é aplicado também às tentativas malsucedidas, evitando que o aviso de erro se repita continuamente.

A revisão 2.1 do firmware foi compilada no Arduino IDE 2.3.10 para a placa “AI Thinker ESP32-CAM”, sem erros, conforme a captura de tela da Figura 49. O relatório do compilador indica que o programa ocupa 500.467 bytes (15%) dos 3.145.728 bytes disponíveis no esquema de partições selecionado e que as variáveis globais ocupam 51.840 bytes (15%) da memória dinâmica, restando 275.840 bytes para variáveis locais. A compilação mostra apenas que o código é sintaticamente válido e cabe na memória do microcontrolador com as bibliotecas instaladas; ela não demonstra o funcionamento do circuito, que depende da montagem e dos ensaios da Tabela 17. As versões do núcleo ESP32 e das bibliotecas utilizadas não constam da captura e devem ser registradas junto ao código-fonte. O código-fonte e uma segunda captura da compilação constam do Apêndice A.

Figura 49 – Compilação da revisão 2.1 do firmware no Arduino IDE 2.3.10 para a placa AI Thinker ESP32-CAM



Fonte: Elaborado pelos autores (2026), captura de tela do Arduino IDE.

4.8 Riscos Técnicos e Medidas de Atenuação

A análise do projeto permitiu identificar riscos que só se materializam na montagem. A Tabela 15 os relaciona com a medida de mitigação já incorporada ou recomendada. A identificação antecipada desses pontos é, ela própria, um resultado do trabalho, pois orienta a montagem futura e reduz o retrabalho.

Tabela 15 – Riscos técnicos identificados e atenuação

Risco	Consequência possível	Atenuação
Compartilhamento do barramento SCCB/I2C	Falha na inicialização da câmera	Diretiva SHARED_SCCB_BUS; alternativa com valor 0; ensaio de reinicializações
Filtro IR-cut da lente OV2640	Imagem noturna escura mesmo com LEDs acesos	Utilizar módulo de câmera NoIR
MOSFET sem nível lógico (ex.: IRFZ44N)	Condução parcial, aquecimento e pouca luz	Adotar IRLZ44N ou AO3400
Nível alto fraco do PCF8574 no gate	Tensão de gate insuficiente com resistor de 100 kΩ	Elevar o resistor para ~470 kΩ ou suprimi-lo; verificar na montagem

GPIO 12 usado como MOSI (pino de boot)	Falha de inicialização se houver nível alto na partida	Não instalar pull-up externo; conferir o nível na energização
GPIO 1 e 3 usados pelo display e pelo ECHO	Serial indisponível; interferência na gravação	Desconectar os fios para gravar; prever conector de programação
Picos de corrente (câmera, SD e IR)	Reinicializações por brownout	Fonte de 5 V e 2 A, capacitor de 470 µF, cabos curtos
DS1307 com pull-ups em 5 V e deriva de relógio	Tensão excessiva no I2C; erro de horário ao longo do tempo	Remover pull-ups do módulo e usar 4,7 kΩ em 3,3 V; considerar DS3231
Reflexão especular e ângulo do HC-SR04	Falhas de detecção ou disparos indevidos	Posicionamento do sensor; ajuste do limiar e de N
Umidade e respingos	Falha de sensores e curto-circuitos	Encapsulamento com proteção; sensor afastado da lâmina de água

Fonte: Elaborado pelos autores (2026).

4.9 Levantamento de Custos de Materiais

Um dos princípios do projeto é a acessibilidade da solução a tutores e clínicas veterinárias. A Tabela 16 lista os componentes necessários à construção do protótipo, com valores unitários aproximados de mercado nacional, adotados como referência inicial e sujeitos à atualização por cotação formal. Em relação à listagem anterior, foram incluídos os itens que o dimensionamento elétrico da Seção 4.2 tornou necessários, como os resistores de *pull-up* do I2C, os capacitores de desacoplamento e o MOSFET de nível lógico, e corrigidas as quantidades de resistores conforme o esquema definitivo.

Tabela 16 – Custo estimado de materiais do protótipo HydraPet Vision v2.0

Item	Componente	Especificação	Qtd.	Unitário	Total
1	Placa ESP32-CAM AI-Thinker	SoC ESP32 dual-core, 4 MB de PSRAM, slot MicroSD	1	R$ 55,00	R$ 55,00
2	Módulo de câmera OV2640 sem filtro IR	Versão NoIR de 2 MP, para visão noturna	1	R$ 28,00	R$ 28,00
3	Display TFT ILI9341 de 2,4"	320 × 240 pixels, interface SPI	1	R$ 48,00	R$ 48,00
4	Expansor PCF8574	Expansor de E/S I2C de 8 bits (0x20)	1	R$ 12,00	R$ 12,00
5	Módulo RTC DS1307	Relógio de tempo real I2C com bateria CR2032	1	R$ 14,00	R$ 14,00
6	Sensor ultrassônico HC-SR04	Faixa de 2 cm a 400 cm, alimentação de 5 V	1	R$ 11,00	R$ 11,00

7	Módulo sensor de luz (LDR/LM393)	Saída digital com ajuste de limiar	1	R$ 8,00	R$ 8,00
8	Cartão MicroSD de 16 GB	Classe 10, formatado em FAT32	1	R$ 25,00	R$ 25,00
9	LEDs infravermelhos 850 nm de 5 mm	Matriz de iluminação noturna	6	R$ 1,20	R$ 7,20
10	MOSFET de nível lógico	IRLZ44N ou AO3400 (chave dos LEDs IR)	1	R$ 3,50	R$ 3,50
11	Botões táteis 12 × 12 mm	Avança, Confirma e Retrocede	3	R$ 1,00	R$ 3,00
12	Resistores de 100 Ω, 1/4 W	Limitadores dos ramos IR (3) e do gate (1)	4	R$ 0,20	R$ 0,80
13	Resistor de 1 kΩ, 1/4 W	Divisor do ECHO (R1)	1	R$ 0,20	R$ 0,20
14	Resistor de 2 kΩ, 1/4 W	Divisor do ECHO (R2)	1	R$ 0,20	R$ 0,20
15	Resistor de descarga do gate	100 kΩ a 470 kΩ (Seção 4.2.4)	1	R$ 0,20	R$ 0,20
16	Resistores de 4,7 kΩ, 1/4 W	Pull-ups do barramento I2C	2	R$ 0,20	R$ 0,40
17	Capacitor eletrolítico de 470 µF	Desacoplamento da alimentação	1	R$ 1,50	R$ 1,50
18	Capacitores cerâmicos de 100 nF	Desacoplamento junto aos módulos	3	R$ 0,30	R$ 0,90
19	Placa perfurada de fenolite	Montagem do circuito auxiliar	1	R$ 12,00	R$ 12,00
20	Conectores e jumpers	Barras de pinos e cabos Dupont	1	R$ 10,00	R$ 10,00
Item	**Componente**	**Especificação**	**Qtd.**	**Unitário**	**Total**
21	Encapsulamento impresso em 3D (PLA)	Cerca de 150 g de filamento	150 g	R$ 0,15 /g	R$ 22,50
22	Fonte de alimentação 5 V / 2 A	Fonte estabilizada com conector adequado	1	R$ 22,00	R$ 22,00
	TOTAL	Custo de materiais do protótipo			R$ 285,40

Fonte: Elaborado pelos autores (2026). Valores aproximados, sujeitos a variação de fornecedor e de data de cotação.

O custo total de materiais resulta em R$ 285,40. Esse valor considera módulos avulsos de prototipagem, o que inclui componentes de aplicação didática (como a placa perfurada e os conectores) que seriam eliminados em um produto industrializado. Reconhece-se que uma estimativa de custo em escala, com placa de circuito impresso dedicada e injeção plástica,

exigiria cotações específicas que não integram o escopo deste trabalho; da mesma forma, não foi realizado levantamento formal de preços de soluções concorrentes, razão pela qual não se estabelece comparação numérica com produtos comerciais. Ainda assim, o uso de módulos de ampla disponibilidade, sem componentes especializados de alto custo, sustenta o atendimento ao requisito RNF03.

4.10 Plano de Validação Experimental

Os resultados deste capítulo são de natureza conceitual e devem ser confirmados com o protótipo montado. A Tabela 17 propõe o roteiro de ensaios para essa etapa, com métricas e critérios de aceitação definidos previamente, de modo a evitar que o limiar de sucesso seja estabelecido depois da observação dos resultados. Os critérios são propostos pelos autores e podem ser ajustados pelo tutor.

Tabela 17 – Roteiro de ensaios para validação física do protótipo

Ensaio	Procedimento	Métrica	Critério de aceitação
E1 – Inicialização	50 reinicializações a frio e a quente com todos os módulos conectados	Reconhecimento de câmera, RTC, expansor e SD	100% dos ciclos
E2 – Exatidão do ultrassom	Alvo plano a 5, 10, 15, 20 e 30 cm, com 30 medidas por distância	Erro médio e desvio-padrão	Erro médio de até 1 cm
E3 – Falsos disparos	100 ciclos de repouso com recipiente cheio e água em movimento, sem animal	Fotos indevidas	Nenhuma ou no máximo 2%
E4 – Detecção	Aproximações repetidas de alvo representativo e, se autorizado, do animal	Taxa de detecção	Pelo menos 95%
E5 – Imagem noturna	Captura a menos de 1 lux nos modos IR_OFF, IR_AUTO e IR_ON	Nitidez, ruído e identificação do animal	Identificação em IR_AUTO
E6 – Latência	Registro do tempo entre a primeira leitura válida e o arquivo gravado	Latência média e máxima	Máximo de 1 s
E7 – Consumo	Medição de corrente nos quatro estados da Tabela de consumo	Corrente por estado	Desvio de até 20% do orçamento
E8 – Persistência	200 capturas seguidas e 20 cortes de energia durante a operação	Arquivos e linhas íntegros	Integridade, exceto o evento em curso

E9 – Operação contínua	7 dias ininterruptos em ambiente doméstico	Reinicializações e deriva do RTC	Sem reinicializações espúrias

Fonte: Elaborado pelos autores (2026).

Os ensaios com animais vivos, ainda que apenas observacionais e sem contato, devem ser precedidos de consulta à Comissão de Ética no Uso de Animais da instituição, de modo a confirmar a necessidade de autorização. Enquanto essa etapa não é cumprida, os ensaios E3, E4 e E5 podem ser conduzidos com alvos e simuladores de aproximação.

4.11 Discussão Geral e Perspectivas de Expansão

A análise conjunta dos resultados indica que a proposta do trabalho é tecnicamente coerente: os requisitos possuem correspondência no firmware, os níveis elétricos foram verificados, o ciclo de detecção ao registro tem duração estimada inferior a um segundo, o consumo cabe folgadamente na fonte prevista e o custo de materiais é compatível com a proposta de uma solução acessível. Ao mesmo tempo, a mesma análise evidenciou pontos que teriam passado despercebidos apenas com a simulação lógica, como a exigência de MOSFET de nível lógico, a fragilidade do resistor de descarga do *gate* e a necessidade de uma câmera sem filtro de infravermelho. Esse achado reforça a importância da verificação analítica como etapa anterior à montagem.

A escolha por uma arquitetura *offline* mostrou-se coerente com o objetivo. Ao manter o processamento e o armazenamento no próprio equipamento, o sistema não depende do roteador doméstico, e as fotografias permanecem no cartão MicroSD sob controle do tutor, sem envio automático a serviços externos, o que reduz a exposição de imagens do ambiente doméstico. A interrupção de hardware para o eco do HC-SR04 e o ciclo cooperativo mantêm a interface responsiva sem recorrer a sistema operacional em tempo real, com menor complexidade de código.

Do ponto de vista da aplicação, é importante delimitar o que o dispositivo mede. O HydraPet Vision registra **eventos** de aproximação ao recipiente e sua evidência fotográfica, e não o volume de água ingerido. A relação entre a frequência de visitas e a saúde urinária ou renal, discutida no Capítulo 2, fundamenta o interesse do registro, mas a interpretação clínica dos dados cabe ao médico-veterinário, e o volume ingerido exigiria recursos adicionais, como a balança sob o recipiente proposta nas melhorias futuras (ROBBINS et al., 2019).

A conectividade do SoC ESP32 permite evoluir o sistema sem alterar o esquema elétrico. O Tabela 18 compara a implementação atual com a expansão prevista para uma versão

3.0, em que o armazenamento local permaneceria como mecanismo principal de segurança contra perda de dados.

Tabela 18 – Roteiro de evolução: versão 2.0 (offline) e versão 3.0 (conectada)

Funcionalidade	Versão 2.0 (projeto atual)	Expansão prevista (versão 3.0)
Armazenamento	CSV e JPEG no MicroSD; parâmetros na NVS	Sincronização com serviço em nuvem (por exemplo, AWS IoT ou TagoIO)
Comunicação	Consulta local no display TFT	Envio dos eventos por MQTT ou HTTP
Alertas	Indicadores visuais na tela inicial	Notificação no smartphone diante de mudança de padrão
Ajuste do relógio	Manual, pelo menu	Sincronização por NTP via Wi-Fi
Visualização de fotos	Galeria no display de 2,4"	Consulta remota por servidor web embarcado ou armazenamento em nuvem
Identificação do animal	Detecção de proximidade (HC-SR04)	Reconhecimento individual por rede neural convolucional

Fonte: Elaborado pelos autores (2026).

4.12 Síntese do Capítulo

Este capítulo apresentou a avaliação conceitual do HydraPet Vision v2.0, realizada por análise do código-fonte, cálculo de engenharia, estimativas de projeto e rastreamento lógico de cenários, uma vez que o dispositivo físico não foi construído. Os principais resultados podem ser resumidos nos seguintes pontos:

a) os treze requisitos definidos possuem mecanismo correspondente no firmware, e os que dependem do desempenho físico foram identificados e encaminhados ao plano de validação;
b) o divisor resistivo do sinal ECHO fornece 3,33 V, dentro dos limites do ESP32 com tolerâncias de ±5%, embora o pior caso combinado (fonte a 5,25 V) chegue a 3,61 V e exija resistores de 1%, e a matriz de seis LEDs infravermelhos drena cerca de 66 mA;
c) a confirmação por quatro leituras consecutivas consome 180 ms, e a latência estimada entre a primeira leitura válida e o registro concluído é inferior a um segundo;
d) o consumo estimado varia de 255 mA a 481 mA a 5 V, com margem de cerca de quatro vezes na fonte de 2 A, mas a operação por baterias exigirá redução do consumo do display;

e) a capacidade de um cartão de 16 GB não é limitante, sendo o tempo de leitura do CSV e o tamanho dos diretórios os pontos a otimizar;
f) a análise identificou riscos de montagem, como o uso de MOSFET sem nível lógico, o resistor de descarga do gate e o filtro de infravermelho da câmera, com as respectivas mitigações; e
g) o custo estimado de materiais é de R$ 285,40, compatível com a proposta de protótipo acessível; a viabilidade comercial não foi avaliada (Seção 4.13).

Conclui-se que o projeto se mostra tecnicamente viável no plano conceitual, e que o passo seguinte, a construção do protótipo e a execução do roteiro de ensaios, transformará as estimativas aqui apresentadas em dados medidos. O Capítulo 5 apresenta as conclusões do trabalho, o cumprimento dos objetivos e as propostas de trabalhos futuros.

4.13 Limitações do Trabalho

Esta seção reúne as limitações que o leitor deve ter em mente ao interpretar os resultados.

Ausência de ensaios. O protótipo físico não foi montado. Taxa de detecção, falsos positivos, qualidade das imagens (inclusive sob infravermelho), latência, consumo e estabilidade elétrica são estimativas ou hipóteses, e não medições. A revisão 2.1 do firmware foi compilada sem erros (Figura 49), mas nem ela nem a versão 2.0 foram executadas em hardware neste trabalho.

Evento não é volume. O dispositivo registra visitas e fotografias, e não o volume de água ingerido. O levantamento com tutores (Seção 2.4.1) indicou interesse em informações como o volume consumido e em saber qual gato bebeu; nenhum dos dois é atendido pela versão atual, que depende de recursos futuros (célula de carga com HX711 e identificação individual).

Presença não é ingestão. O sensor ultrassônico detecta um objeto dentro da faixa, e não a ingestão de água. Uma visita pode, portanto, corresponder a um gato que apenas se aproxima, a outro animal ou a uma pessoa, e, em casas com vários gatos, não há identificação individual. A frequência de visitas não substitui a medição do consumo nem a avaliação veterinária.

Ultrassom e felinos. O HC-SR04 emite pulsos de 40 kHz. A audição do gato doméstico alcança o ultrassom, com limite superior relatado de cerca de 85 kHz (HEFFNER; HEFFNER, 1985), de modo que a emissão pode ser percebida e incomodar o animal. Este trabalho não avaliou essa reação, e o requisito RNF04 permanece pendente de observação do comportamento.

Infravermelho. LEDs de 850 nm emitem um brilho avermelhado tênue, perceptível em ambiente totalmente escuro; LEDs de 940 nm seriam invisíveis, mas a sensibilidade do sensor OV2640 é menor nessa faixa. Além disso, a maioria dos módulos OV2640 traz filtro de corte de infravermelho, que pode inviabilizar a imagem noturna sem a troca da lente ou do módulo (Seção 4.4.2).

Viabilidade econômica e amostra. O custo de R$ 285,40 refere-se aos materiais de um protótipo e não é preço de venda. Não houve levantamento de preços de produtos concorrentes nem comparação entre o custo e a disposição a pagar declarada na pesquisa, de modo que a viabilidade econômica ou comercial não está demonstrada. A pesquisa com 98 tutores usou amostra de conveniência, obtida por formulário online, e não permite generalização.

Parâmetros não calibrados. Os parâmetros de visita (8 s de intervalo e seis fotos por visita), o limiar de 15 cm e o filtro de quatro leituras são escolhas de projeto, ainda não calibradas com animais reais.

Privacidade. As fotografias do ambiente doméstico ficam no cartão MicroSD sem criptografia; quem tiver acesso físico ao cartão tem acesso às imagens.

5. CONCLUSÃO

O desenvolvimento, a codificação em firmware e a verificação analítica do protótipo HydraPet Vision, em sua versão 2.0, indicaram, no plano conceitual, a viabilidade técnica de uma solução autônoma e de custo estimado reduzido para o registro fotográfico de visitas ao bebedouro felino. A evolução da primeira prova de conceito para uma versão mais estruturada do sistema permitiu resolver limitações relacionadas ao uso da placa ESP32-CAM, principalmente em função da quantidade reduzida de GPIOs disponíveis e da necessidade de integrar diferentes periféricos em uma mesma plataforma embarcada.

A problemática que motivou este trabalho está relacionada à dificuldade encontrada por tutores e médicos-veterinários em acompanhar, de forma contínua e não invasiva, o comportamento de ingestão de água dos gatos em ambientes domésticos. Para lidar com essa questão, o projeto reuniu conhecimentos de sistemas embarcados, eletrônica digital, visão computacional, Internet das Coisas, medicina veterinária preventiva e manufatura aditiva. A proposta foi utilizar a tecnologia para automatizar o registro lógico dos eventos de acesso ao recipiente com água, criando informações que possam contribuir para o acompanhamento do comportamento hídrico dos animais e auxiliar na identificação de alterações que mereçam atenção.

Nesse contexto, o HydraPet Vision 2.0 foi estruturado como uma estação local de monitoramento, com código-fonte integralmente desenvolvido em C++ para realizar a detecção da aproximação do animal, confirmar o evento por meio de leituras consecutivas, controlar a iluminação infravermelha, programar o registro fotográfico, armazenar as imagens e manter um histórico lógico dos eventos. É fundamental reiterar que a validação do projeto foi conduzida por meio de análise estática do código-fonte, rastreamento lógico de cenários operacionais, cálculos de engenharia elétrica e simulação conceitual, visto que o protótipo físico não foi fisicamente montado em bancada. Essa abordagem permitiu verificar a coerência lógica do firmware e da arquitetura, mas não demonstra robustez em operação real, o que depende da montagem e dos ensaios da Tabela 17.

Essa integração entre software, especificação de sensores, estratégia de armazenamento e interface lógica permitiu transformar a proposta teórica em um sistema organizado e adequado para futura montagem; o uso contínuo no ambiente doméstico depende dos ensaios.

5.1 Síntese das Conquistas e Resultados da Versão 2.0

Uma das principais soluções adotadas na versão 2.0 e validada em nível de firmware foi a utilização do expansor de portas digitais PCF8574 no barramento I2C. Essa implementação permitiu especificar e programar a ampliação da quantidade de entradas e saídas disponíveis sem comprometer as conexões utilizadas pela câmera, pelo display TFT e pelo cartão MicroSD. Por meio do expansor, especificou-se a integração dos três botões da interface, o acionamento lógico do sensor ultrassônico, a leitura do módulo LDR e o controle do circuito de iluminação infravermelha. O RTC DS1307 também foi integrado ao mesmo barramento I2C, formando uma arquitetura de software coesa que aproveita os recursos limitados da ESP32-CAM.

Outra evolução importante ocorreu no sistema de sensoriamento. A leitura do HC-SR04 foi estruturada de maneira não bloqueante em código, utilizando interrupção por mudança de estado no sinal de eco conectado ao GPIO 3. Com essa abordagem, o processamento do sistema não fica interrompido durante a medição da distância, permitindo que outras funções continuem sendo executadas concorrentemente. Para aumentar a confiabilidade da detecção, também foi incorporada a confirmação por leituras consecutivas no firmware, reduzindo a ocorrência de falsos positivos provocados por ruídos acústicos ou reflexões ocasionais.

O sistema de iluminação infravermelha também foi integrado ao fluxo lógico de captura das imagens. A especificação do uso de seis LEDs de 850 nm, acionados por meio de um transistor MOSFET, possibilita a programação de registros em condições de baixa luminosidade. Como essa faixa de infravermelho não é percebida pelo olho humano, ela permite complementar a iluminação durante a captura simulada sem utilizar iluminação branca convencional. O controle da iluminação foi programado para ocorrer de acordo com as condições identificadas pelo sensor LDR e pelas configurações selecionadas no sistema, permitindo o funcionamento automático ou manual.

A interface homem-máquina (IHM) programada para o display TFT ILI9341 de 2,4 polegadas contribuiu para projetar um equipamento independente. Por meio do código desenvolvido, o usuário poderá consultar os registros armazenados, navegar pela galeria de fotografias, ajustar a data e a hora do RTC, alterar os parâmetros de funcionamento e verificar informações relacionadas ao estado do sistema, dispensando o uso de computadores para a operação cotidiana.

A persistência dos dados foi outro aspecto consolidado na arquitetura de software da versão 2.0. As configurações do sistema foram programadas para serem armazenadas na

memória não volátil por meio da biblioteca Preferences, permitindo que os parâmetros definidos pelo usuário sejam recuperados após uma interrupção da alimentação. Os registros dos eventos foram estruturados para gravação no arquivo registros.csv, enquanto as fotografias seguem organizadas em subpastas por data em /fotos do cartão MicroSD. Essa estrutura de software favorece a rastreabilidade dos dados coletados conceitualmente.

O desenvolvimento do projeto mecânico do gabinete por meio de impressão 3D complementou a especificação do protótipo. A fabricação pelo processo FDM foi concebida para acomodar os componentes eletrônicos em uma estrutura única, protegendo a montagem contra respingos e facilitando sua futura utilização prática.

5.2 Trabalhos e Evoluções Futuras

Embora a versão 2.0 tenha alcançado uma estrutura de firmware coerente e verificada de forma analítica, o projeto foi concebido de maneira modular, permitindo a incorporação de novos recursos e a posterior montagem física em trabalhos futuros. As possibilidades de evolução concentram-se na montagem prática do hardware em bancada, na execução do plano de ensaios experimentais, na identificação individual dos animais, na obtenção de medidas quantitativas de consumo e na disponibilização dos dados por meio de serviços em nuvem e aplicativos móveis.

Uma das possibilidades de maior interesse para a continuidade do projeto é a montagem física do circuito em placa de circuito impresso (PCI) dedicada e a posterior incorporação de técnicas de Visão Computacional e Aprendizado de Máquina diretamente no dispositivo, utilizando conceitos de Edge AI. Essa evolução seria relevante em residências com múltiplos gatos, permitindo associar os registros fotográficos a cada animal específico por meio de Redes Neurais Convolucionais (CNNs) otimizadas com TensorFlow Lite for Microcontrollers.

Outra evolução natural consiste na montagem mecânica da célula de carga associada ao conversor HX711 na base do reservatório, viabilizando a obtenção de medidas quantitativas de volume hídrico consumido. Adicionalmente, a montagem de um circuito de alimentação de emergência com baterias de íon-lítio (18650) e um módulo BMS garantirá autonomia contra quedas de energia.

Por fim, aproveitando a capacidade de rádio integrada do SoC ESP32, futuras versões poderão implementar conectividade sem fio (Wi-Fi e Bluetooth Low Energy) para o envio de registros via MQTT/HTTP para plataformas de IoT (AWS IoT, TagoIO) e notificações em

aplicativos móveis (Android e iOS), transformando a especificação conceitual verificada analiticamente neste trabalho em um produto comercial completo.

5.3 Considerações finais

A trajetória de desenvolvimento do HydraPet Vision 2.0 demonstrou como os conhecimentos de sistemas embarcados e engenharia de firmware podem ser aplicados à especificação de soluções inovadoras voltadas ao monitoramento e ao bem-estar animal. O projeto superou desafios importantes relacionados à escassez de GPIOs, à estruturação de interrupções de hardware, ao gerenciamento de barramentos I2C compartilhados e à persistência de dados em arquivos locais e memória NVS.

A principal contribuição deste trabalho reside na especificação rigorosa e na codificação completa de um sistema integrado capaz de unir detecção de proximidade não bloqueante, confirmação lógica de eventos, captura e gerenciamento de imagens, persistência tabular em CSV e interface gráfica local. Como a validação foi conduzida integralmente por meio de simulação lógica e análise de firmware, dada a opção metodológica de não montar o protótipo físico, o trabalho entrega uma base de código testada conceitualmente e pronta para ser implementada em hardware em etapas futuras.

Dessa forma, o HydraPet Vision 2.0 representa uma contribuição acadêmica para o campo dos sistemas embarcados e da tecnologia aplicada à medicina veterinária preventiva, abrindo caminhos consistentes para a futura fabricação do protótipo físico e para o desenvolvimento de novas soluções voltadas à qualidade de vida dos animais de estimação.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAFRUIT INDUSTRIES. **Adafruit GFX Library**. [S. l.]: Adafruit, [s.d.]. Disponível em: https://github.com/adafruit/Adafruit-GFX-Library. Acesso em: 24 out. 2026.

ADAFRUIT INDUSTRIES. **Adafruit ILI9341 library**. [S. l.]: Adafruit, [s.d.]. Disponível em: https://github.com/adafruit/Adafruit_ILI9341. Acesso em: 24 out. 2026.

ANALOG DEVICES. **DS1307: 64 x 8, serial, I²C real-time clock**. Datasheet. Rev. 5. Wilmington: Analog Devices, 2015. Disponível em: https://www.analog.com/en/products/ds1307.html. Acesso em: 24 out. 2026.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação: referências: elaboração. Rio de Janeiro: ABNT, 2018.

BAHGA, A.; MADISETTI, V. **Internet of things**: a hands-on approach. [S. l.]: VPT, 2014.

BODMER. **TJpg_Decoder**: Arduino library for decoding JPEG images. [S. l.]: GitHub, [s.d.]. Disponível em: https://github.com/Bodmer/TJpg_Decoder. Acesso em: 24 out. 2026.

BUCKLEY, C. M. et al. Effect of dietary water intake on urinary output, specific gravity and relative supersaturation for calcium oxalate and struvite in the cat. **British Journal of Nutrition**, v. 106, supl. 1, p. S128-S130, 2011.

CASTILLO-ARCEO, O. E.; RENTERIA-FLORES, R. U.; SANTANA-MANCILLA, P. C. Design and development of a smart pet feeder with IoT and deep learning. **Engineering Proceedings**, v. 82, n. 1, art. 63, 2024. DOI: 10.3390/ecsa-11-20487. Disponível em: https://www.mdpi.com/2673-4591/82/1/63. Acesso em: 24 out. 2026.

CHAN, E. N. **FatFs: generic FAT filesystem module**. [S. l.]: ChaN, [s.d.]. Disponível em: http://elm-chan.org/fsw/ff/. Acesso em: 24 out. 2026.

DIBARTOLA, S. P. **Fluid, electrolyte, and acid-base disorders in small animal practice**. 4. ed. St. Louis: Elsevier Saunders, 2012.

DRISCOLL, C. A. et al. The Near Eastern origin of cat domestication. **Science**, v. 317, n. 5837, p. 519-523, 2007.

ELECFREAKS. **Ultrasonic ranging module HC-SR04**. Datasheet. [S. l.]: Elecfreaks, [s.d.]. Disponível em: https://mischianti.org/wp-content/uploads/2025/11/Datasheet-HCSR04-classic.pdf (cópia hospedada por terceiro). Acesso em: 24 out. 2026.

ESPRESSIF SYSTEMS. **ESP32 series datasheet**. [S. l.]: Espressif Systems, [s.d.a]. Disponível em: https://www.espressif.com/. Acesso em: 24 out. 2026.

ESPRESSIF SYSTEMS. **esp32-camera: driver for OV2640, OV3660, OV5640, OV7670 and OV7725 image sensors**. [S. l.]: Espressif Systems, [s.d.b]. Disponível em: https://github.com/espressif/esp32-camera. Acesso em: 24 out. 2026.

FITRIANI, A. **Alat pemantauan dan pengontrol pemberian pakan kucing dengan sensor HC-SR04 e camera ESP32-CAM, berbasis Internet of Things (IoT)**. 2024. Trabalho de conclusão de curso – Universitas Islam Riau, [s. l.], 2024. Disponível em: https://repository.uir.ac.id/28703. Acesso em: 24 out. 2026.

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. 4. ed. New York: Pearson, 2018.

HEFFNER, R. S.; HEFFNER, H. E. Hearing range of the domestic cat. **Hearing Research**, v. 19, n. 1, p. 85-88, 1985.

ILI TECHNOLOGY CORP. **ILI9341**: a-Si TFT LCD single chip driver 240RGBx320 resolution and 262K color. Datasheet. Version 1.02. [S. l.]: ILI Technology, [s.d.]. Disponível em: https://academy.cba.mit.edu/classes/output_devices/TFT/ILI9341.pdf. Acesso em: 24 out. 2026.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T T.81**: information technology: digital compression and coding of continuous-tone still images: requirements and guidelines. Geneva: ITU, 1992. Disponível em: https://www.itu.int/rec/T-REC-T.81-199209-I/en. Acesso em: 24 out. 2026.

KOTWAL, J. et al. Deep-feed: an Internet of things-enabled smart feeding system for pets powered by deep learning. **IAES International Journal of Robotics and Automation**, v. 14, n. 2, p. 227-236, 2025. DOI: 10.11591/ijra.v14i2.pp227-236. Acesso em: 24 out. 2026.

LEE, E. A.; SESHIA, S. A. **Introduction to embedded systems**: a cyber-physical systems approach. 2. ed. Cambridge: MIT Press, 2017.

LITTLE, S. E. **The cat**: clinical medicine and management. St. Louis: Saunders Elsevier, 2011.

MARINO, C. L. et al. Prevalence and classification of chronic kidney disease in cats randomly selected from four age groups and in cats recruited for degenerative joint disease studies. **Journal of Feline Medicine and Surgery**, v. 16, n. 6, p. 465-472, 2014. DOI: 10.1177/1098612X13511446.

MICROSOFT CORPORATION. **FAT**: general overview of on-disk format. Hardware White Paper. Version 1.02. [S. l.]: Microsoft, 1999. Cópia consultada em: https://www.pjrc.com/docs/specs/disk/fat_spec.html. Acesso em: 24 out. 2026.

MONK, S. **Programming Arduino**: getting started with sketches. 2. ed. New York: McGraw-Hill Education, 2016.

NICOLAE, S.; CODREANU, I. Ethological study regarding the dipsic behaviour changes during the gestation period in domestic cats. **Scientific Works. Series C. Veterinary Medicine**, v. 70, n. 2, 2024. Disponível em: https://www.veterinarymedicinejournal.usamv.ro/pdf/2024/issue_2/Art2.pdf. Acesso em: 24 out. 2026.

NXP SEMICONDUCTORS. **PCF8574; PCF8574A: remote 8-bit I/O expander for I²C-bus with interrupt**. Product data sheet. Rev. 5. Eindhoven: NXP, 2013. Disponível em: https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf. Acesso em: 24 out. 2026.

NXP SEMICONDUCTORS. **UM10204: I²C-bus specification and user manual**. Rev. 7.0. Eindhoven: NXP, 2021. Disponível em: https://www.nxp.com/docs/en/user-guide/UM10204.pdf. Acesso em: 24 out. 2026.

OMNIVISION TECHNOLOGIES. **OV2640 color CMOS UXGA (2.0 megapixel) CameraChip with OmniPixel2 technology**. Advanced information preliminary datasheet. Version 1.6. [S. l.]: OmniVision, 2006. Disponível em: https://files.waveshare.com/wiki/common/OV2640DS_en.pdf (cópia hospedada por terceiro). Acesso em: 24 out. 2026.

PETKIT. **EverSweet Ultra with Camera pet water fountain**. [S. l.]: Petkit, [s.d.]. Disponível em: https://www.petkit.com/products/eversweet-ultra-with-camera-pet-water-fountain. Acesso em: 24 out. 2026.

PETLIBRO. **Dockstream 2 smart fountain**. [S. l.]: Petlibro, [s.d.]. Disponível em: https://petlibro.com/products/dockstream-app-monitoring-water-fountain. Acesso em: 24 out. 2026.

POLZIN, D. J. Chronic kidney disease in small animals. **Veterinary Clinics of North America: Small Animal Practice**, v. 41, n. 1, p. 15-30, 2011.

POLZIN, D. J. Evidence-based step-wise approach to managing chronic kidney disease in dogs and cats. **Journal of Veterinary Emergency and Critical Care**, v. 23, n. 2, p. 205-215, 2013. DOI: 10.1111/vec.12034.

REIS, P. M.; JUNG, S.; ARISTOFF, J. M.; STOCKER, R. How cats lap: water uptake by Felis catus. **Science**, v. 330, n. 6008, p. 1231-1234, 2010. DOI: 10.1126/science.1195421.

ROBBINS, M. T. et al. Quantified water intake in laboratory cats from still, free-falling and circulating water bowls, and its effects on selected urinary parameters. **Journal of Feline Medicine and Surgery**, v. 21, n. 8, p. 682-690, 2019. DOI: 10.1177/1098612X18803753.

SPARKES, A. H. et al. ISFM consensus guidelines on the diagnosis and management of feline chronic kidney disease. **Journal of Feline Medicine and Surgery**, v. 18, n. 3, p. 219-239, 2016. DOI: 10.1177/1098612X16631234.

STELLA, J.; CRONEY, C.; BUFFINGTON, T. Effects of stressors on the behavior and physiology of domestic cats. **Applied Animal Behaviour Science**, v. 143, n. 2-4, p. 157-163, 2013. DOI: 10.1016/j.applanim.2012.10.014.

SURE PETCARE. **Felaqua Connect**. [S. l.]: Sure Petcare, [s.d.]. Disponível em: https://www.surepetcare.com/pt-pt/felaqua-connect. Acesso em: 24 out. 2026.

TEXAS INSTRUMENTS. **LM393B, LM2903B, LM193, LM293, LM393 and LM2903 dual comparators**. Datasheet SLCS005, Rev. AH. Dallas: Texas Instruments, 2025. Disponível em: https://www.ti.com/document-viewer/LM393/datasheet. Acesso em: 24 out. 2026.

TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. **Sistemas digitais**: princípios e aplicações. 10. ed. São Paulo: Pearson Prentice Hall, 2007.

UNTIVEROS-PEÑALOZA, H.; CASTRO-VARGAS, C. IoT-based smart pet feeding system with dynamic network configuration using ESP32, WiFiManager and Blynk 2.0. **Ingénierie des Systèmes d'Information**, v. 30, n. 11, p. 2881-2889, 2025. DOI: 10.18280/isi.301106. Acesso em: 24 out. 2026.

ZANGHI, B. M.; GERHEART, L.; GARDNER, C. L. Effects of a nutrient-enriched water on water intake and indices of hydration in healthy domestic cats fed a dry kibble diet. **American Journal of Veterinary Research**, v. 79, n. 7, p. 733-744, 2018.

ZORAN, D. L.; BUFFINGTON, C. A. T. Effects of nutrition choices and lifestyle changes on the well-being of cats, a carnivore that has moved indoors. **Journal of the American Veterinary Medical Association**, v. 239, n. 5, p. 596-606, 2011.

APÊNDICE A – CÓDIGO-FONTE DO FIRMWARE E EVIDÊNCIA DE COMPILAÇÃO

Este apêndice reúne as informações necessárias para obter, compilar e verificar o código-fonte do firmware do HydraPet Vision, revisão 2.1.0, descrito na Seção 3.19. O arquivo completo é disponibilizado em repositório público no GitHub, e o apêndice registra o ambiente de compilação utilizado e a evidência de que o código foi compilado sem erros.

A.1 Disponibilização do código-fonte

O código-fonte está publicado no GitHub, de onde pode ser baixado, consultado e reproduzido por qualquer leitor deste trabalho. O endereço do repositório e o link para download devem ser inseridos pelos autores nos campos destacados abaixo, antes da entrega do documento.

Repositório no GitHub: https://github.com/GbMarguti/HydroPet_Vision?tab=readme-ov-file
Link para download (arquivo .zip ou release): [inserir aqui o link para download]
Arquivo principal: HydraPet_Vision_v2_1.ino (revisão 2.1.0)
Data de acesso: [inserir a data de acesso]

Recomenda-se publicar, junto ao arquivo .ino, o esquema elétrico e a lista de bibliotecas com as versões utilizadas, de modo que a compilação possa ser reproduzida por terceiros.

A.2 Ambiente de compilação

Para reproduzir a compilação, devem ser seguidos os passos abaixo, conforme o cabeçalho do próprio código-fonte:

a) instalar o Arduino IDE 2.x e, no Gerenciador de Placas, o núcleo “esp32”, da Espressif (série 3.x);
b) instalar, no Gerenciador de Bibliotecas, as bibliotecas Adafruit GFX Library, Adafruit ILI9341 (que instala a Adafruit BusIO), RTClib e TJpg_Decoder; as demais (Wire, SPI, Preferences, FS, SD_MMC e esp_camera) são nativas do núcleo ESP32;
c) selecionar a placa “AI Thinker ESP32-CAM”, com PSRAM habilitada (Enabled), esquema de partição “Huge APP (3MB No OTA/1MB SPIFFS)” e Core Debug Level “None”, pois o GPIO1 é usado pelo display;

d) manter a pasta do sketch com o nome HydraPet_Vision_v2_1 e acionar o comando Verificar;

e) para gravar na placa, desconectar os fios do GPIO1 e do GPIO3 (DC e ECHO), ligar o GPIO0 ao GND e, após a gravação, remover o jumper e reiniciar a placa.

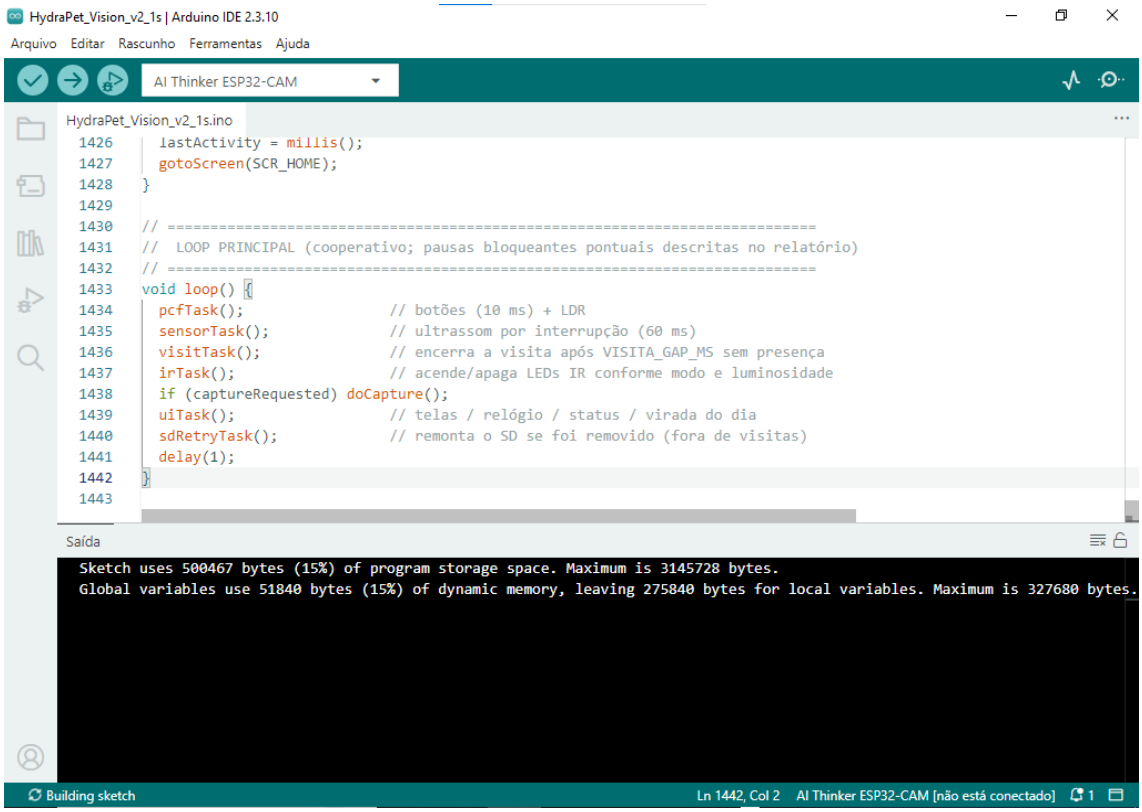
As versões utilizadas na compilação devem ser registradas nos campos destacados:

Arduino IDE: 2.3.10
Núcleo esp32 (Espressif): [inserir versão]
Adafruit GFX Library: [inserir versão]
Adafruit ILI9341: [inserir versão]
Adafruit BusIO: [inserir versão]
RTClib: [inserir versão]
TJpg_Decoder: [inserir versão]

A.3 Evidência de compilação

A Figura 50 apresenta a captura de tela do Arduino IDE 2.3.10 com o resultado da compilação do firmware para a placa AI Thinker ESP32-CAM. O relatório do compilador indica que o programa ocupa 500.467 bytes (15%) dos 3.145.728 bytes disponíveis para programa e que as variáveis globais ocupam 51.840 bytes (15%) dos 327.680 bytes de memória dinâmica, restando 275.840 bytes para variáveis locais. A captura exibe ainda o trecho final do código, com a função loop() e as tarefas do ciclo principal.

Figura 50 – Compilação do firmware do HydraPet Vision no Arduino IDE 2.3.10, com o relatório de uso de memória



Fonte: Elaborado pelos autores (2026), captura de tela do Arduino IDE.

Essa compilação demonstra que o código é sintaticamente válido e cabe na memória do microcontrolador com as bibliotecas instaladas, mas não comprova o funcionamento do circuito, que depende da montagem do protótipo e dos ensaios previstos na Seção 4.10.