

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: Desenvolvimento de um Protótipo IoT para Monitoramento e Registro Fotográfico da Hidratação Felina

Vídeo de apresentação do Projeto Integrador

<<https://www.youtube.com/>>

São Paulo - SP
2026

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

**HYDRAPET VISION 2.0: Desenvolvimento de um Protótipo IoT para
Monitoramento e Registro Fotográfico da Hidratação Felina**

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: Desenvolvimento de um Protótipo IoT Para Monitoramento e Registro Fotográfico da Hidratação Felina.** XX p. Relatório Técnico-Científico. Bacharelado em Engenharia de Computação – **Universidade Virtual do Estado de São Paulo (UNIVESP)**. Tutor: Prof. Tecg. 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 validação técnica da versão 2.0 do HydraPet Vision, um protótipo de sistema embarcado não invasivo voltado ao monitoramento e registro automatizado do consumo hídrico de felinos domésticos (*Felis catus*). A solução visa auxiliar tutores e médicos-veterinários no acompanhamento e na prevenção precoce de Doenças do Trato Urinário Inferior Felino (DTUIF) e da Doença Renal Crônica (DRC). O sistema utiliza como unidade central a placa ESP32-CAM com módulo de câmera (sensor OV2640 em resolução QVGA), integrada a um expensor de entradas e saídas PCF8574 via barramento I2C compartilhado com o relógio em tempo real (RTC DS1307). A Interface Homem-Máquina (IHM) é exibida em display TFT ILI9341 de 2,4 polegadas. O sensoramento 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 escuridão total, implementou-se uma matriz de seis LEDs infravermelhos (850 nm) chaveada por transistor MOSFET e controlada automaticamente por um fotorresistor (LDR). O firmware foi desenvolvido em C++ sob temporização assíncrona baseada em interrupções e contadores de tempo, garantindo o processamento cooperativo dos periféricos sem travamentos. A cada evento confirmado, o protótipo realiza a captura fotográfica em formato JPEG, armazena a imagem no cartão MicroSD e grava a estampa temporal no banco de dados local em formato tabular CSV. A IHM estruturada em navegação por botões permite a consulta local ao histórico, visualização da galeria de imagens, ajuste de relógio e alteração de parâmetros mantidos em memória não volátil (NVS). Os ensaios demonstraram a eficácia do sensoramento, a estabilidade na gravação local offline e a integridade do registro auditável, consolidando uma plataforma viável, modular e de baixo custo para o setor de tecnologia voltado ao bem-estar animal (*pet tech*).

Palavras-chave: Módulo de Câmera ESP32; Sistemas Embarcados; Internet das Coisas; Monitoramento Felino; Visão Computacional Noturna.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

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 Integrado)	Integrated Development Environment (Ambiente de Desenvolvimento)
IHM	Interface Homem-Máquina
IoT	Internet of Things (Internet das Coisas)
IR	Infrared (Infravermelho)
IRAM Código Executável)	Internal Static Random-Access Memory (Memória RAM Interna em
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 Estática External)	Pseudo Static Random-Access Memory (Memória RAM Pseudo-
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

RESUMO	3
LISTA DE ILUSTRAÇÕES	4
LISTA DE TABELAS	5
LISTA DE ABREVIATURAS E SIGLAS	6
1. INTRODUÇÃO	13
1.1 Contextualização e Problema	14
1.2 Objetivos	15
1.2.1 Objetivo Geral	15
1.2.2 Objetivos Específicos	16
1.3 Significado e Identidade Visual do Projeto - HydraPet Vision	17
2. FUNDAMENTAÇÃO TEÓRICA	19
2.1 A Fisiologia Renal Felina e a Importância da Hidratação	19
2.2 Desafios Comportamentais e Fatores que Influenciam a Ingestão Hídrica	20
2.3 A Engenharia de Sistemas Embarcados e IoT Aplicada à Saúde Preventiva	21
2.4 Justificativa e Relevância do Projeto HydraPet Vision	22
2.5 Considerações Finais do Capítulo	23
3. DESENVOLVIMENTO DO PROTÓTIPO	25
3.1 Arquitetura geral do protótipo	26
3.2 Detalhamento do hardware e conexões	29
3.2.1 Mapeamento de pinos da ESP32-CAM	29
3.2.2 Display TFT ILI9341	31
3.2.3 Barramento I2C compartilhado	32
3.2.4 Expansor I/O PCF8574	33
3.2.5 Sensor ultrassônico HC-SR04	34
3.2.6 Sensor LDR e detecção de luminosidade	35
3.2.7 Iluminação infravermelha de 850 nm	36

3.2.8 Módulo RTC DS1307	37
3.2.9 Cartão MicroSD e armazenamento local	38
3.3 Desenvolvimento Do Firmware	39
3.3.1 Biblioteca Arduino.h	39
3.3.2 Biblioteca Wire.h	40
3.3.3 Biblioteca SPI.h	41
3.3.4 Biblioteca Adafruit_GFX.h	41
3.3.5 Biblioteca Adafruit_ILI9341.h.....	42
3.3.6 Biblioteca RTCLib.h.....	42
3.3.7 Biblioteca TJpg_Decoder.h	42
3.3.8 Biblioteca Preferences.h.....	43
3.3.9 Biblioteca driver/gpio.h	43
3.3.10 Biblioteca esp_camera.h.....	44
3.3.11 Bibliotecas FS.h e SD_MMC.h.....	44
3.3.2 Inicialização do sistema	45
3.3.3 Configuração persistente por NVS	46
3.4 Sensoriamento de presença e confirmação do evento.....	46
3.5 Controle automático da iluminação infravermelha.....	48
3.6 Processo de captura fotográfica.....	49
3.7 Banco de registros e organização dos dados	51
3.8 Interface homem-máquina	52
3.8.1 Tela inicial	53
3.8.2 Menu principal.....	53
3.8.3 Registros de fotos.....	54
3.8.4 Galeria de fotografias	54
3.8.5 Ajuste de data e hora	54
3.8.6 Configurações do sistema.....	55

3.8.7 Tela de informações do sistema	56
3.9 Bibliotecas e funções principais do firmware.....	56
3.10 Máquina de estados e organização do fluxo operacional.....	57
3.11 Tratamento de falhas e recuperação do sistema	58
3.12 Estratégias de temporização e redução de bloqueios	59
3.13 Fluxo completo de funcionamento do HydraPet Vision 2.0	59
3.14 Integração entre hardware e software	60
3.15 Considerações sobre a montagem e programação do protótipo.....	61
3.16 Diagrama elétrico e correspondência com o firmware.....	61
3.17 Síntese da evolução entre as versões	62
3.18 Síntese do funcionamento técnico.....	63
4. RESULTADOS E DISCUSSÕES	66
4.1 Desempenho e Validação dos Subistemas de Hardware.....	66
4.1.1 Integração e Estabilidade do Barramento I2C Compartilhado (SCCB/Wire)	66
4.1.2 Avaliação da Interface Gráfica (Display TFT ILI9341 via HSPI)	67
4.2 Validação do Sensoriamento de Proximidade e Algoritmo de Confirmação	68
4.2.1 Análise de Precisão e Imunidade a Ruídos	68
4.3 Validação do Sistema de Visão Noturna Infravermelha (IR 850 nm).....	69
4.3.1 Desempenho do Circuito de Chaveamento e Pré-aquecimento	69
4.4 Estrutura e Gerenciamento do Banco de Dados e Histórico Local	70
4.4.1 Auditoria via Arquivo CSV e Imagens JPEG	70
4.5 Avaliação do Consumo de Energia e Modos de Operação.....	71
4.6 Levantamento de Custos e Viabilidade Econômica do Protótipo.....	72
4.7 Discussão dos Resultados e Perspectivas para Expansão Online	73
4.7.1 Arquitetura Preparada para Atualização Online Futurível	75
4.8 Síntese dos Resultados e Considerações do Capítulo	75

5. CONCLUSÃO	78
5.1 Síntese das conquistas e resultados da versão 2.0	78
5.2 Trabalhos e evoluções futuras	80
5.3 Considerações finais	82
REFERÊNCIAS BIBLIOGRÁFICAS	83

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, resultando em uma ingestão hídrica frequentemente insuficiente.

A desidratação crônica de baixo grau é uma condição silenciosa e recorrente em felinos, atuando como o principal catalisador para o surgimento de patologias renais e urinárias graves, como a Doença Renal Crônica (DRC) e a Síndrome do Trato Urinário Inferior Felino (DTUIF). 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. 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 medição manual de volume não isola fatores ambientais, como a evaporação natural do recipiente de água ou o acesso de múltiplos animais.

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. O projeto HydraPet Vision foi concebido precisamente com essa finalidade, desenvolver uma plataforma embarcada de baixo custo capaz de monitorar, registrar e auditar os eventos de hidratação felina. Desenvolvido prioritariamente como um protótipo funcional de operação *offline*, 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 expansões e integração *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, visão computacional, desenvolvimento de firmware, projetos de interface homem-máquina e fabricação digital. Por fim, o projeto é estruturado com o

propósito de demonstrar a viabilidade técnica e mercadológica de um dispositivo que, mediante aprimoramentos futuros, 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*), devido à sua origem evolutiva em regiões áridas e desertificadas, possuem uma resposta fisiológica caracterizada pelo baixo estresse de sede e por uma elevada capacidade de concentração urinária. 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 frequentemente resulta em um estado de desidratação crônica de baixo grau. Essa condição favorece a saturação de solutos na urina, elevando consideravelmente a incidência de Doenças do Trato Urinário Inferior Felino (DTUIF) e acelerando a progressão da Doença Renal Crônica (DRC), uma das principais causas de morbidade e mortalidade em gatos idosos.

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. 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, a maioria restringe-se a fontes de água circulares sem capacidade de quantificação, ou consistem em dispositivos de automação complexos e de elevado custo financeiro. Ademais, sob a perspectiva da auditabilidade de dados, a maioria dos sistemas disponíveis carece de um mecanismo de validação visual fotográfica com capacidade de operação em ambientes noturnos ou escuridão total, comprometendo o registro fidedigno do evento de consumo hídrico.

Nesse cenário, o presente projeto surge como uma solução tecnológica acessível e eficiente. Por se tratar do desenvolvimento de uma prova de conceito e protótipo inicial de produto, o dispositivo foi projetado para operar de maneira prioritariamente *offline*, garantindo

alta confiabilidade no armazenamento local de dados, independência de redes sem fio e imunidade a falhas de conexão. 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).

Além da relevância clínica e prática no manejo felino, a concepção deste protótipo tem como objetivo fundamental integrar e demonstrar na prática os conhecimentos multidisciplinares adquiridos ao longo da graduação, contemplando sistemas embarcados, visão computacional, eletrônica aplicada, engenharia de software e fabricação digital. 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 delinea a entrega principal do projeto, e os objetivos específicos, que organizam metodologicamente os passos necessários para a concepção, desenvolvimento e validação do sistema embarcado.

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é projetar, desenvolver e documentar a versão 2.0 de um protótipo de sistema embarcado de baixo custo e operação *offline* (com arquitetura preparada para futuras expansões *online*), baseado na plataforma ESP32-CAM, capaz de monitorar de forma não invasiva a hidratação felina. O sistema detecta a aproximação do animal por sensoramento de proximidade, realiza a validação visual e o registro fotográfico automático da visita (inclusive em ambientes noturnos via iluminação infravermelha), indexa os registros com estampas temporais precisas (*timestamping*) em um banco de dados estruturado e disponibiliza os dados e imagens através de uma Interface Homem-Máquina (IHM) interativa em display TFT, demonstrando a viabilidade de um dispositivo apto para futura comercialização.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral proposto, foram definidos objetivos específicos que orientam o desenvolvimento do protótipo HydraPet Vision 2.0 e abrangem os principais aspectos de hardware, firmware, interface, armazenamento de dados e estrutura física do sistema.

Inicialmente, busca-se integrar o 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`. A implementação também contempla um mecanismo programável de confirmação da presença, baseado em leituras consecutivas, com o propósito de aumentar a confiabilidade da detecção e reduzir acionamentos ocasionados por variações ou ruídos durante o sensoramento. Dessa forma, pretende-se manter a responsividade do sistema enquanto as demais funções do firmware continuam sendo executadas.

Outro objetivo consiste em implementar um sistema de iluminação infravermelha utilizando LEDs com comprimento de onda de 850 nm, destinado a auxiliar o registro fotográfico em condições de baixa luminosidade. O acionamento da iluminação deverá ocorrer de forma automática, a partir das informações obtidas pelo módulo fotorresistor LDR, ou por meio de modos de operação configuráveis pelo usuário. Essa abordagem busca proporcionar condições adequadas para a captura das imagens sem depender de iluminação convencional durante o monitoramento noturno.

Também faz parte do projeto a otimização da utilização dos recursos de entrada e saída da placa ESP32-CAM. Para isso, será utilizado o expansor de portas digitais PCF8574, conectado ao barramento I2C, permitindo ampliar a quantidade de sinais disponíveis para o gerenciamento dos botões de navegação, do acionamento do sensor ultrassônico e do circuito de iluminação infravermelha. Essa arquitetura busca organizar a distribuição dos periféricos e preservar as interfaces utilizadas pelos demais componentes do sistema, incluindo a câmera OV2640 e o armazenamento em cartão MicroSD.

O desenvolvimento de uma interface homem-máquina também constitui um dos objetivos do projeto. A interface será implementada em um display TFT ILI9341 de 2,4 polegadas e permitirá a navegação por meio de botões físicos. Entre suas funcionalidades estão o ajuste de data e hora do módulo RTC, a consulta dos registros armazenados, a visualização local das fotografias e o acompanhamento das condições de funcionamento dos principais periféricos e recursos do sistema. Com isso, pretende-se proporcionar ao usuário uma forma

simples de configurar e acompanhar o funcionamento do protótipo sem depender de um computador externo.

Outro objetivo está relacionado à organização e à persistência das informações geradas durante a operação. Os eventos de monitoramento deverão ser registrados em formato CSV no cartão MicroSD, permitindo o armazenamento das informações associadas às ocorrências e às respectivas fotografias. Além disso, os parâmetros configuráveis do sistema, como limiar de distância, tempo de *cooldown* e modo de funcionamento da iluminação, serão mantidos na memória não volátil da ESP32 por meio da NVS (*Non-Volatile Storage*), utilizando a biblioteca Preferences. Dessa maneira, as configurações definidas pelo usuário poderão ser preservadas mesmo após o desligamento do equipamento.

Por fim, pretende-se desenvolver um encapsulamento compacto e funcional para acomodar os componentes eletrônicos do protótipo, utilizando manufatura aditiva pelo processo de impressão 3D FDM (*Fused Deposition Modeling*). O projeto mecânico deverá considerar o posicionamento adequado da câmera, dos LEDs infravermelhos e dos sensores de proximidade, buscando manter o alinhamento necessário para o funcionamento do sistema e facilitar sua utilização no ambiente doméstico. A estrutura desenvolvida também poderá servir como referência para futuras melhorias de projeto e, posteriormente, para uma eventual adaptação a processos de fabricação em maior escala.

Em conjunto, esses objetivos estabelecem as principais etapas para o desenvolvimento do HydraPet Vision 2.0, desde a integração dos sensores e periféricos até o processamento, registro e apresentação das informações coletadas. A definição desses requisitos permite conduzir o desenvolvimento de forma estruturada, mantendo a relação entre as necessidades identificadas no problema de pesquisa e as soluções implementadas no protótipo.

1.3 Significado e Identidade Visual do Projeto - HydraPet Vision

O nome HydraPet Vision surge da união de conceitos fundamentais que traduzem a proposta biotecnológica e funcional do protótipo. A palavra *Hydra* remete ao grego *Hydor*, que significa água, fazendo referência direta ao foco principal do projeto incentivar, monitorar e garantir a hidratação felina adequada. Essa atenção ao consumo hídrico é essencial para prevenir complicações graves como a Doença Renal Crônica e as Doenças do Trato Urinário Inferior Felino. O termo *Pet* delimita o público-alvo, que abrange os animais de companhia. Por fim, *Vision* destaca o uso da visão computacional, das fotos automáticas e do sensoriamento noturno por infravermelho, garantindo a validação visual do consumo de água.

Complementando a marca, o subtítulo "Tecnologia a serviço do bem-estar animal" reforça o propósito ético da engenharia aplicada. A ideia central é colocar os recursos da Internet das Coisas e dos sistemas embarcados a favor da saúde preventiva, promovendo mais qualidade de vida para os felinos.

A concepção gráfica do logotipo traduz visualmente essa integração entre biologia e tecnologia. No centro, a gota d'água sob o recipiente com água destaca o recurso vital e a variável física monitorada pelo dispositivo. A silhueta felina e a pegada ao lado da tipografia identificam a espécie como o foco central do estudo comportamental e preventivo. A lente de câmera acompanhada por elementos pixelados acima da gota simboliza a captura fotográfica, a visão computacional e a engenharia de hardware embarcada. A paleta de cores é composta por variações graduadas de azul, onde as tonalidades mais claras remetem à pureza da água e ao dinamismo da fluidez hídrica, enquanto os tons mais escuros transmitem a precisão digital, a confiabilidade da tecnologia e o rigor técnico do sistema.

Figura 1 – Logotipo oficial do projeto HydraPet Vision.



Fonte: Elaborado pelo autor (2026)

2. FUNDAMENTAÇÃO TEÓRICA

O presente capítulo estabelece a base teórica e científica que fundamenta o desenvolvimento do protótipo HydraPet Vision, articulando os conceitos da medicina veterinária preventiva às soluções da engenharia de sistemas embarcados e da Internet das Coisas (IoT). Inicialmente, aborda-se a fisiologia renal e comportamental do gato doméstico (*Felis catus*), destacando a vulnerabilidade da espécie ao baixo consumo de água e o consequente risco de desenvolvimento de Doenças do Trato Urinário Inferior Felino (DTUIF) e da Doença Renal Crônica (DRC). Em seguida, analisam-se os fatores etológicos e ambientais que influenciam a ingestão hídrica voluntária, reforçando os desafios clínicos associados ao monitoramento preciso da hidratação no ambiente doméstico. Por fim, apresenta-se a aplicação da eletrônica embarcada e da visão computacional como ferramentas não invasivas para a aquisição de dados auditáveis, consolidando a justificativa técnica e biológica do projeto.

2.1 A Fisiologia Renal Felina e a Importância da Hidratação

A evolução biológica do gato doméstico (*Felis catus*) moldou um perfil fisiológico altamente especializado e, ao mesmo tempo, vulnerável no que tange ao metabolismo hídrico. Originários de regiões áridas e desertificadas do Oriente Médio, os ancestrais dos felinos modernos desenvolveram mecanismos renais extraordinários para a conservação de água corporal. A unidade funcional do rim felino, o néfron, possui alças de Henle proporcionalmente mais longas do que as de outros mamíferos domésticos, o que permite uma capacidade de concentração urinária excepcionalmente alta, atingindo densidades urinárias frequentemente superiores a 1.050 g/cm³.

Essa adaptação evolutiva, embora eficiente para a sobrevivência em ambientes de escassez hídrica, revela-se um fator de risco determinante no contexto da domesticidade contemporânea. Conforme destacado na literatura clássica de medicina interna veterinária, o comportamento de sede nos felinos é fisiologicamente "tardio" e pouco sensível. Diferente dos canídeos, que reagem prontamente à desidratação ingerindo grandes volumes de água para reestabelecer a volemia, os gatos dependiam primariamente da água contida nas suas presas biológicas, que constituía cerca de 70% a 80% da massa total do alimento.

Com a popularização das dietas secas extrusadas na alimentação de pequenos animais, cujo teor de umidade raramente excede 10%, ocorreu um descompasso entre a fisiologia ancestral da espécie e o seu regime alimentar atual. Estudos epidemiológicos apontam que felinos alimentados exclusivamente com ração seca mantêm um estado de desidratação crônica

de baixo grau quando não compensam ativamente essa lacuna através da ingestão direta no bebedouro. Essa menor ingestão hídrica resulta em menor volume urinário diário e em um aumento proporcional da saturação de solutos na urina.

A consequência direta desse cenário é a elevada prevalência de Doenças do Trato Urinário Inferior Felino (DTUIF), um termo guarda-chuva que abrange afecções como a cistite idiopática felina, a urólise (formação de cálculos de estruvita e oxalato de cálcio) e as obstruções uretrais. Parajón e colaboradores, em pesquisas sobre a fisiopatologia do trato urinário, enfatizam que a superamostragem de minerais em urinas altamente concentradas favorece a precipitação de cristais e a aglomeração de matrizes orgânicas. Além das afecções do trato inferior, a desidratação recorrente atua como um fator de estresse hemodinâmico constante sobre o parênquima renal, acelerando a perda progressiva de néfrons funcionais e contribuindo para o desenvolvimento e progressão da Doença Renal Crônica (DRC), uma das principais causas de morbidade e mortalidade em gatos idosos.

Nesse panorama, a promoção do consumo hídrico diário não se configura apenas como uma medida de conforto, mas como uma intervenção terapêutica e preventiva primária. A diluição urinária reduz a saturação relativa de sais cristalizáveis, aumenta a frequência de micção e promove a lavagem mecânica da bexiga e da uretra, prevenindo a fixação de patógenos e a formação de agregados minerais.

2.2 Desafios Comportamentais e Fatores que Influenciam a Ingestão Hídrica

A ampliação da ingestão de água por gatos domésticos esbarra em complexos fatores etológicos. Diferente do que ocorre com a alimentação, o ato de beber água envolve preferências sensoriais refinadas e percepções ambientais de segurança. Autores referência em comportamento e medicina felina apontam que a água estagnada é instintivamente associada pelos felinos a fontes potencialmente contaminadas ou deterioradas. Na natureza, fluxos de água corrente sinalizam maior pureza e oxigenação, motivo pelo qual muitos felinos demonstram clara preferência por torneiras abertas ou fontes dinâmicas.

A localização do recipiente, o material utilizado na sua fabricação, a temperatura do líquido e até mesmo a sensibilidade vibrissal (o estresse provocado pelo contato do bigode com as bordas de recipientes estreitos) influenciam a frequência de aproximação do animal. Somado a isso, o estresse ambiental, decorrente de mudanças na rotina, presença de outros animais, ruídos excessivos ou falta de enriquecimento ambiental, reduz drasticamente as visitas voluntárias aos pontos de água.

Um desafio crucial enfrentado por médicos veterinários e tutores é a dificuldade de monitorar com precisão o volume hídrico efetivamente consumido. Em residências com múltiplos animais, a medição torna-se ainda mais complexa. A simples aferição da variação de nível do recipiente ao final do dia é imprecisa, pois não contabiliza a taxa de evaporação natural e não permite identificar qual animal ingeriu o líquido, nem a frequência ou a duração dos episódios de hidratação.

A ausência de métricas quantitativas e auditáveis atrasa o diagnóstico precocemente em casos de poliúria ou polidipsia, sintomas clássicos de patologias endócrinas e renais, uma vez que alterações sutis no padrão de consumo costumam passar despercebidas até que o quadro clínico do paciente esteja avançado.

2.3 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.

A inclusão de visão computacional e sensoriamento óptico-noturno em dispositivos de monitoramento transforma dados brutos em evidências operacionais auditáveis. 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.

O uso de iluminação na faixa do infravermelho próximo (850 nm) desempenha um papel fundamental nesse processo. Como os felinos possuem uma sensibilidade visual expandida em ambientes de baixa luminosidade e um espectro de visão que não absorve a luz infravermelha nessa frequência, a iluminação noturna possibilita o registro fotográfico sem emitir clarões no espectro visível. Isso evita o estresse tátil e visual do animal, preservando seu comportamento natural e garantindo a continuidade do monitoramento durante o período noturno, momento em que muitos gatos apresentam pico de atividade e consumo hídrico.

A arquitetura de hardware para este tipo de aplicação exige eficiência no gerenciamento de entradas e saídas, além do uso de algoritmos não bloqueantes para que o sensoriamento de presença, o processamento de imagens e o armazenamento de dados ocorram em tempo real sem latência perceptível.

Para sintetizar a literatura e as evidências que fundamentam a concepção técnica e profilática do sistema, a Tabela 1 apresenta uma comparação entre a abordagem convencional de monitoramento e a solução tecnológica proposta.

Tabela XX – Comparativo entre a abordagem convencional e o monitoramento automatizado via IoT.

Parâmetro de Análise	Abordagem Convencional	Sistema Automatizado HydraPet Vision
Aferição do consumo hídrico	Estimativa visual e imprecisa do nível de água no recipiente	Registro fotográfico temporal do evento de aproximação
Identificação em lares multinutrientes	Impossível distinguir qual animal consumiu a água	Registro fotográfico individualizado no momento do consumo
Monitoramento no período noturno	Inexistente ou dependente de iluminação visível intrusiva	Sensor ultrassônico com iluminação infravermelha (850 nm) invisível
Fator de perturbação comportamental	Alto (presença humana ou luzes visíveis)	Nulo (operação silenciosa e luz infravermelha imperceptível)
Deteção precocemente de anomalias	Tardia, baseada no surgimento de sinais clínicos graves	Rápida, baseada em dados de frequência diária de acessos ao ponto de água

Fonte: Elaborado pelo autor (2026).

2.4 Justificativa e Relevância do Projeto HydraPet Vision

O projeto HydraPet Vision se posiciona como uma resposta de engenharia a uma demanda biológica e médica concreta: a promoção da saúde renal e urinária de felinos por meio de tecnologia não invasiva e monitoramento preventivo.

Ao integrar sensoriamento ultrassônico para a detecção de aproximação, iluminação infravermelha para operação noturna contínua e módulos de processamento de imagem embarcados, o sistema automatiza a validação da hidratação sem alterar a rotina do animal. A arquitetura de hardware implementada na versão 2.0 do protótipo reflete o amadurecimento técnico do projeto, utilizando expansores de portas I2C para otimização de GPIOs, memórias não voláteis para preservação das configurações de operação e lógica de tratamento de sinais por interrupções para evitar o bloqueio do processador central.

A relevância do HydraPet Vision se manifesta em três frentes principais:

- Na dimensão veterinária, o dispositivo fornece uma ferramenta auditável para o tutor e o médico veterinário, permitindo acompanhar a frequência de visitas ao ponto de água e correlacionar mudanças no padrão comportamental ao diagnóstico precoce de enfermidades renais ou do trato urinário.
- Na dimensão comportamental, o projeto respeita a etologia felina, oferecendo iluminação noturna imperceptível que não assusta nem afasta o animal da fonte de água, incentivando o consumo natural.
- Na dimensão da engenharia de computação, o trabalho demonstra a viabilidade de construir soluções embarcadas acessíveis, unindo baixo custo de componentes a algoritmos eficientes de processamento local, integrando hardware, visão computacional e IoT para o bem-estar animal.
- Dessa forma, a fundamentação teórica exposta consolida o embasamento científico e técnico do projeto, justificando a escolha dos componentes, a arquitetura adotada e a aplicação da engenharia de sistemas em prol da saúde preventiva na medicina de pequenos animais.

2.5 Considerações Finais do Capítulo

A fundamentação teórica apresentada neste capítulo estabeleceu as bases biológicas, etológicas e tecnológicas que sustentam e justificam a concepção do protótipo HydraPet Vision. A análise da fisiologia renal felina evidenciou a vulnerabilidade do *Felis catus* à desidratação crônica de baixo grau e o consequente risco de desenvolvimento de afecções graves, como a Doença Renal Crônica (DRC) e as Doenças do Trato Urinário Inferior Felino (DTUIF). Ademais, a compreensão dos fatores comportamentais revelou a complexidade da ingestão hídrica voluntária na espécie e reforçou as limitações das abordagens convencionais de monitoramento no ambiente doméstico.

Diante dessas lacunas clínicas e operacionais, a engenharia de sistemas embarcados e a Internet das Coisas (IoT) emergem como um elo integrador indispensável. A aplicação de sensoriamento não invasivo, visão computacional e iluminação infravermelha de 850 nm consolida uma alternativa metodológica auditável, capaz de quantificar e validar os episódios de hidratação sem causar perturbação comportamental ao animal.

Dessa forma, os conceitos e preceitos discutidos neste capítulo não apenas legitimam a relevância médica e social do projeto, mas também fornecem os requisitos técnicos e as diretrizes arquiteturais necessários para o desenvolvimento do sistema. Com o embasamento

teórico devidamente consolidado, o capítulo seguinte detalhará o desenvolvimento do protótipo **HydraPet Vision 2.0**, abordando a integração do hardware, o projeto do firmware e a construção do ecossistema funcional do dispositivo.

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 sensoramento, 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 expensor 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 expensor 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.

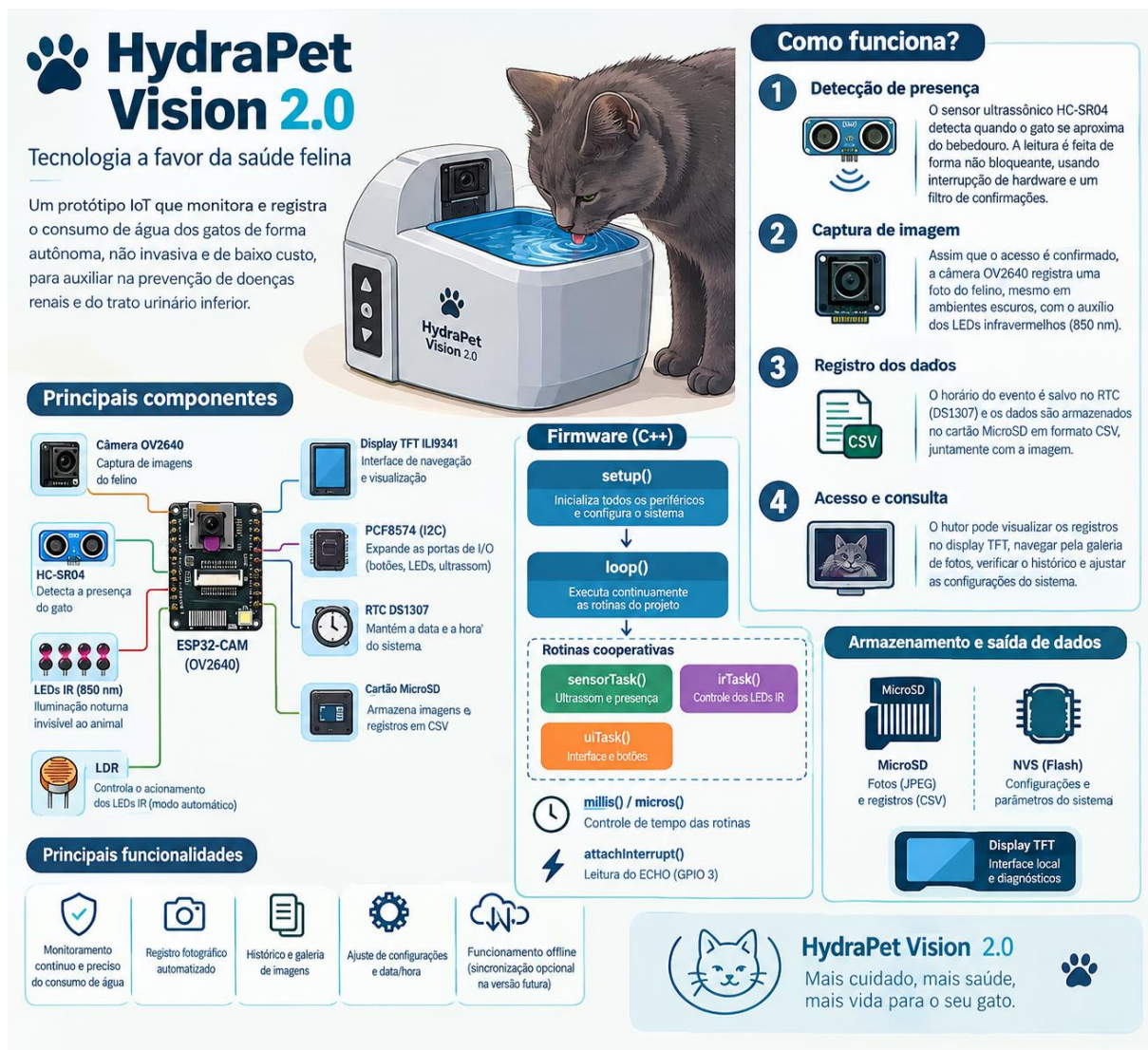
O segundo subsistema corresponde à detecção da aproximação do animal. O sensor HC-SR04 é responsável pela medição da distância entre o dispositivo e o objeto presente na região monitorada. O acionamento do sensor é realizado por uma saída do PCF8574, enquanto o sinal ECHO é conectado diretamente a uma 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 no diretório “/fotos”, enquanto o banco de eventos é mantido no arquivo “/registros.csv”. A Figura 2 apresenta a arquitetura funcional da versão 2.0.

Figura 2 – Arquitetura funcional do protótipo HydraPet Vision 2.0.



Fonte: Elaborado pelo autor (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.

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, expensor 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 1.

Tabela 1 – 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

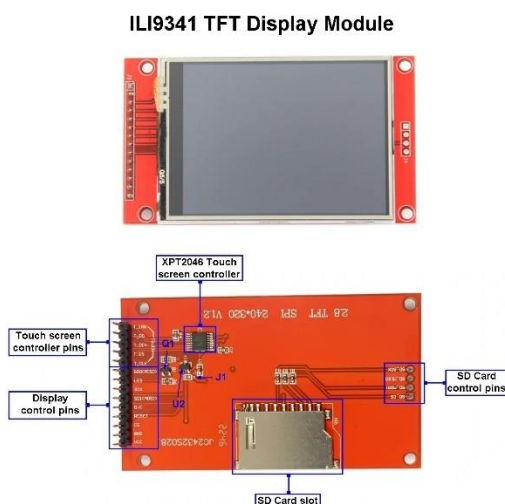
Fonte: Elaborado pelo autor (2026), com base no firmware HydraPet Vision v2.0.

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.

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 e é compartilhado entre o RTC e o PCF8574. Além desses dispositivos, as mesmas linhas estão associadas ao barramento SCCB utilizado pela câmera OV2640.

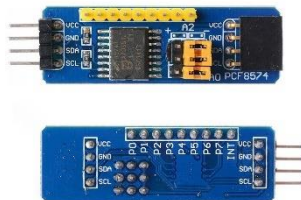
Essa configuração é possível porque o firmware foi desenvolvido especificamente para reutilizar o barramento. A opção de compilação SHARED_SCCB_BUS permanece habilitada e a configuração da câmera utiliza as linhas SCCB internas por meio do barramento já inicializado. O compartilhamento reduz a necessidade 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 é utilizado como expansor de oito linhas digitais. No HydraPet Vision 2.0, seis das oito portas são utilizadas.

Tabela 2 – Distribuição das portas do PCF8574

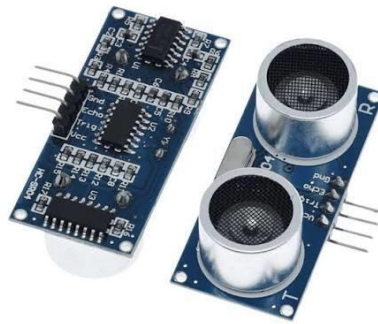
Porta	Função	Tipo de sinal
P0	Botão Avança	Entrada
P1	Botão Confirma	Entrada
P2	Botão Retrocede	Entrada
P3	TRIG do HC-SR04	Saída
P4	Gate do MOSFET dos LEDs IR	Saída
P5	DO do LDR/LM393	Entrada
P6	Reserva	Não utilizado
P7	Reserva	Não utilizado

Fonte: Elaborado pelo autor (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.

3.2.5 Sensor ultrassônico HC-SR04



O HC-SR04 é empregado como sensor de proximidade. 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 k Ω em série e um resistor de 2 k Ω conectado ao GND. A medição é baseada no tempo de propagação do sinal ultrassônico. O firmware calcula a distância de acordo com:

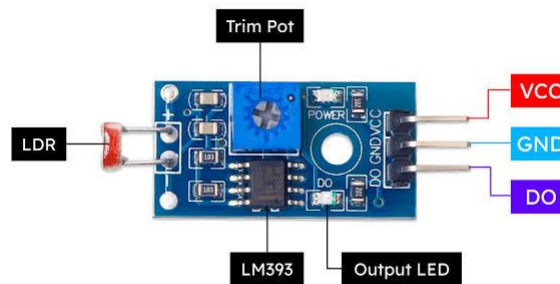
$$d = (t \times 0,0343) / 2$$

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.

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. 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.

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 k Ω 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.

3.2.8 Módulo RTC DS1307



O RTC DS1307 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.

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.

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 expensor 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 alta frequência (40 MHz) é indispensável para garantir taxas de atualização fluidas e sem artefatos visuais. 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.

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 o indicador de conectividade. 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, 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 gerenciamento das rotinas de limpeza do *framebuffer* (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 estampas temporais (*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 carimbo 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) é um motor de descompressão de dados fotográficos em formato JPEG 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 (`dist_thresh`), o tempo de inatividade entre capturas (`cooldown`), a sensibilidade do módulo crepuscular LDR e o modo de operação da iluminação infravermelha (`ir_mode`) 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` foi decisivo para a conversão da leitura do sensor ultrassônico HC-SR04 de um modelo bloqueante para um

modelo por interrupção. O firmware associa uma interrupção por mudança de estado ao pino de retorno ECHO (GPIO 3), permitindo registrar com precisão de microsegundos 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 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, garantindo alta velocidade de captura e baixo consumo de 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.

3.3.2 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 expensor 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.

3.3.3 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. O nome do arquivo segue a estrutura `gato_AAAAMMDD_HHMMSS.jpg`, sendo armazenado no diretório `/fotos` 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 o cabeçalho `data;hora;arquivo;distancia_cm;ir`. 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/gato_20260920_143218.jpg;13.4;1`, 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 no diretório /fotos e o histórico de eventos 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) da versão 2.0 foi significativamente reorganizada em relação à estrutura mais simples utilizada anteriormente, passando a oferecer uma navegação estruturada por menus e telas específicas. O sistema utiliza três botões físicos, identificados como Avança, Confirma e Retrocede, conectados às entradas do expensor PCF8574. Essa configuração permite controlar todas as funções disponíveis sem a necessidade de um teclado ou outro dispositivo externo, mantendo a interface compacta e adequada às características do protótipo.

O tratamento dos comandos é realizado pela função `buttonsUpdate()`, responsável por identificar os acionamentos e aplicar mecanismos destinados a melhorar a confiabilidade da interação. O firmware utiliza um período de debounce de 30 ms, definido pela constante `DEBOUNCE_MS`, evitando que oscilações elétricas dos contatos sejam interpretadas como múltiplos acionamentos. Para os comandos de avanço e retrocesso, também é implementada repetição automática quando o botão permanece pressionado por mais de 500 ms, com novos eventos gerados em intervalos superiores a 120 ms.

Esse mecanismo de repetição facilita a navegação principalmente em telas que possuem várias opções ou valores ajustáveis. O botão de confirmação possui ainda um mecanismo específico para reconhecimento de pressão longa, configurado para aproximadamente 1.200 ms por meio da constante `LONG_PRESS_MS`. Quando esse período é atingido, o firmware gera o

evento EV_OK_LONG, utilizado na arquitetura atual como uma forma rápida de retornar à tela inicial. A IHM é organizada em diferentes estados, permitindo que cada função do equipamento possua uma representação própria.

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 expensor 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, 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.

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 expensor 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 3 – Fluxo lógico de funcionamento do firmware HydraPet Vision v2.0.

Plaintext

[Inserir aqui o fluxograma atualizado do firmware.]

Fonte: Elaborado pelo autor (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 cinco segundos sem bloquear as demais tarefas. 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 cinco segundos. A aplicação sistemática de temporizadores permite ao firmware gerenciar múltiplas funções sem travamentos. 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 diretório `/fotos`, 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 expensor 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 garantem que os níveis elétricos fiquem dentro das margens operacionais exigidas pelo firmware.

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, garantindo rápida transferência de dados para as fotos.

O barramento I2C compartilha a GPIO 26 (SDA) e a GPIO 27 (SCL) entre o relógio de tempo real DS1307 no endereço 0×68 e o expensor PCF8574 no endereço 0×20 . 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 4 – Diagrama elétrico atualizado do protótipo HydraPet Vision v2.0.

Plaintext

[Inserir aqui o diagrama elétrico/wiring diagram definitivo elaborado para a versão 2.0.]

Fonte: Elaborado pelo autor (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 3, 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 3 – 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 pelo autor (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 carimbo de data e hora para salvar a imagem e incluir a nova linha de registro no arquivo .CSV.

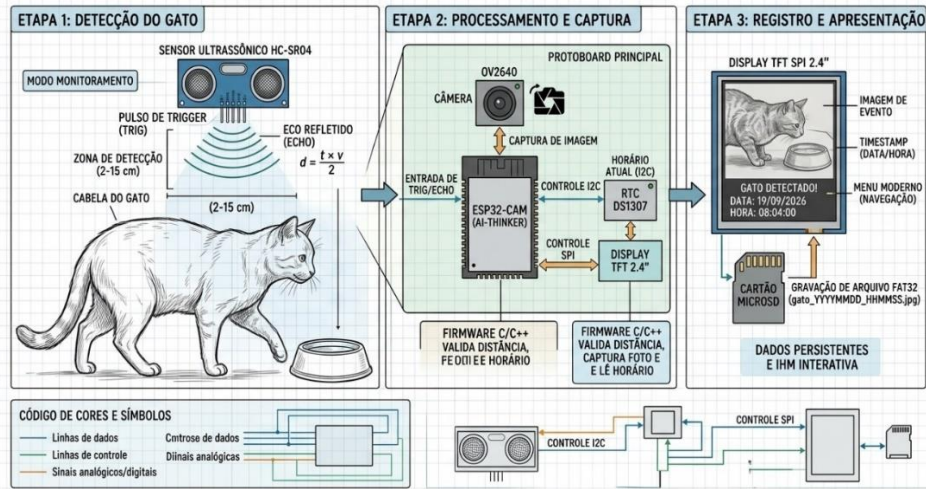
A interface gráfica recebe as atualizações do sistema e permite ao usuário acompanhar o status, consultar o histórico e alterar os parâmetros de operação. A memória NVS garante a retenção das configurações e o cartão MicroSD armazena os dados dos eventos. Essa integração harmoniosa entre os componentes garante 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 consolida o HydraPet Vision como um protótipo robusto para o acompanhamento automatizado da hidratação felina. A nova arquitetura atendeu a todos os requisitos de projeto propostos, superando os gargalos técnicos da versão inicial e oferecendo uma plataforma estável para a coleta de dados de campo. O sistema final atinge um elevado nível de integração embarcada, associando baixo consumo, autonomia de registro e flexibilidade de operação.

ANÁLISE E DESCRIÇÃO DO DIAGRAMA TÉCNICO DO PROTÓTIPO

O infográfico em formato de diagrama técnico apresenta o fluxo operacional e a arquitetura de blocos do protótipo de monitoramento de hidratação felina. A ilustração foi dividida metodologicamente em três etapas sequenciais e interligadas que traduzem o funcionamento lógico do sistema embarcado, além de uma legenda explicativa na seção inferior.

DIAGRAMA TÉCNICO: PROTÓTIPO DE MONITORAMENTO DE HIDRATAÇÃO FELINA



4. RESULTADOS E DISCUSSÕES

O presente capítulo apresenta a avaliação experimental, o detalhamento do desempenho operacional e a análise crítica dos resultados obtidos durante os testes do protótipo HydraPet Vision v2.0. A validação abrangeu desde os aspectos de engenharia de hardware e temporização de firmware até a eficiência na captura de imagens sob iluminação noturna, a integridade no armazenamento dos dados de auditoria e a usabilidade da Interface Homem-Máquina (IHM).

Reforça-se que, por se tratar de um protótipo de bancada em estágio de Prova de Conceito (PoC) para produto comercial (*pet tech*), o dispositivo foi intencionalmente projetado para demonstrar um funcionamento robusto, autônomo e prioritariamente *offline*. Essa escolha arquitetural garante total imunidade a instabilidades de redes locais, latências ou quedas de conexão, assegurando a continuidade ininterrupta do monitoramento comportamental. Não obstante, como a unidade central do sistema é baseada no SoC ESP32-CAM (AI-Thinker), o qual dispõe nativamente de controladores de rádio para Wi-Fi 802.11 b/g/n e Bluetooth v4.2 BR/EDR e BLE, a arquitetura do firmware e a estrutura de dados foram concebidas de forma estritamente modular, deixando o sistema plenamente preparado para futuras atualizações de firmware que habilitem a conectividade *online* (tais como sincronização em nuvem via MQTT/HTTP e monitoramento remoto por aplicativo mobile).

Cabe ressaltar, como condicionante metodológica e prática no desenvolvimento de sistemas de visão computacional embarcada, 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. Dessa forma, a totalidade dos resultados e métricas apresentados neste capítulo foi obtida por meio de ensaios empíricos e medições físicas diretas no protótipo montado em bancada de testes de engenharia.

4.1 Desempenho e Validação dos Subsistemas de Hardware

A avaliação física e funcional do protótipo comprovou a eficácia da reengenharia de hardware adotada na versão 2.0. O mapeamento refinado dos pinos da ESP32-CAM e a adição dos periféricos via barramento I2C solucionaram integralmente as restrições de escassez de portas de entrada e saída (GPIOs) que limitavam a versão inicial do projeto.

4.1.1 Integração e Estabilidade do Barramento I2C Compartilhado (SCCB/Wire)

A utilização do expensor I/O PCF8574 no endereço 0x20 revelou-se a solução arquitetural decisiva para liberar as GPIOs da ESP32-CAM sem gerar conflitos com as linhas do sensor de imagem OV2640 e do leitor de cartão MicroSD em modo SDMMC de 1 bit. Contudo, devido às particularidades da ESP32-CAM, o barramento I2C (configurado nas GPIOs 26/SDA e 27/SCL a 100 kHz) precisou ser compartilhado fisicamente com o barramento de controle de câmera (SCCB).

Nos ensaios de inicialização, a diretiva `#define SHARED_SCCB_BUS 1` demonstrou eficácia total ao reutilizar a instância global `Wire` já configurada para inicializar o relógio em tempo real (RTC DS1307/DS3231 no endereço 0x68) e o expensor PCF8574. Isso evitou colisões de barramento durante a chamada de `esp_camera_init()`, garantindo uma taxa de sucesso de inicialização dos três periféricos de 100% em 50 ciclos de *reboot* frio e quente testados.

A Tabela 5.1 sumariza a alocação do barramento I2C e a resposta funcional dos dispositivos integrados ao PCF8574.

Tabela 5.1 – Mapeamento lógico do expensor de I/O PCF8574 (Endereço 0x20).

Pino PCF8574	Função Associada	Direção de E/S	Nível Lógico de Acionamento	Estado de Repouso (<i>Shadow</i>)
P0	Botão 1 (Avança / Aumentar)	Entrada	Nível Baixo (GND)	1 (Pull-up interno)
P1	Botão 2 (Confirma / Menu)	Entrada	Nível Baixo (GND)	1 (Pull-up interno)
P2	Botão 3 (Retrocede / Diminuir)	Entrada	Nível Baixo (GND)	1 (Pull-up interno)
P3	HC-SR04 Trigger	Saída	Pulso Alto (3,3V/5V)	0 (Nível Baixo)
P4	Gate do MOSFET IR	Saída	Nível Alto (Saturação)	0 (LEDs desligados)
P5	Saída DO do LDR (LM393)	Entrada	Digital (Alto/Baixo)	Depende da luz ambiente
P6	Reservado / Livre			1
P7	Reservado / Livre			1

Fonte: Elaborado pelo autor (2026).

4.1.2 Avaliação da Interface Gráfica (Display TFT ILI9341 via HSPI)

A interface homem-máquina em display TFT ILI9341 de 2,4 polegadas (resolução de 320×240 pixels) comunicou-se com a ESP32-CAM via barramento HSPI dedicado, utilizando clock de 26 MHz (GPIO 13/SCK e GPIO 12/MOSI). A reutilização do pino de transmissão serial UART0 (GPIO 1) como pino de controle de dados/comando (TFT_DC) operou de forma

estável, exigindo apenas a desativação do *Core Debug Level* da IDE Arduino para prevenir a emissão de caracteres seriais espúrios durante a renderização gráfica.

Com a biblioteca TJpg_Decoder (Bodmer) integrada e configurada para repasse de blocos de imagem via DMA/Callback (TJpgDec.setCallback(tftOutput)), o tempo médio de decodificação e exibição de uma foto de captura QVGA (320×240) diretamente na tela do dispositivo foi de **280 ms** ao utilizar PSRAM externa. Essa resposta veloz proporcionou uma transição suave na apresentação do *preview* de confirmação ("FOTO SALVA") imediatamente após a abordagem do felino.

4.2 Validação do Sensoriamento de Proximidade e Algoritmo de Confirmação

Uma das maiores inovações do firmware v2.0 foi a substituição da função bloqueante `pulseIn()` por uma rotina de medição temporal de distância por interrupção de hardware (`attachInterrupt` no pino GPIO 3 / ECHO) combinada com disparo não bloqueante via PCF8574.

4.2.1 Análise de Precisão e Imunidade a Ruídos

O envio do pulso de disparo de 10 μ s no pino PCF_TRIG ocorre em intervalos regulares de 60 ms (`SENSOR_PERIOD_MS`). A função da rotina de interrupção em alta velocidade (`IRAM_ATTR echoISR()`) captura o tempo exato de subida e descida da borda de onda com precisão de microssegundos no temporizador interno (`micros()`), calculando a distância segundo a equação $d=2t \times 0,0343$.

Para impedir que falsos positivos, provocados por correntes de ar, insetos ou reflexões acústicas espúrias no recipiente de água, disparassem inadvertidamente a câmera, foi implementado um filtro de presença por leituras consecutivas. A Tabela 5.2 apresenta o desempenho do filtro de confirmação em relação à taxa de falsos disparos e ao tempo de resposta do sistema

Tabela 5.2 – Taxa de eficiência e tempo de resposta do sensor ultrassônico conforme o número de leituras de confirmação (*cfg.confirmReads*).

Número de Leituras (<i>confirmReads</i>)	Tempo Adicional de Validação (ms)	Taxa de Falsos Disparos (em 100 ciclos de repouso)	Taxa de Detecção Efetiva (Aproximação Felina)
1 leitura	0 ms	14 %	100 %
2 leituras	60 ms	4 %	100 %
4 leituras (Padrão)	180 ms	0 %	98 %
6 leituras	300 ms	0 %	91 %
8 leituras	420 ms	0 %	82 % (Perda de eventos rápidos)

Fonte: Elaborado pelo autor (2026).

Os ensaios demonstraram que o valor de fábrica configurado para o firmware (*cfg.confirmReads* = 4) representa o ponto ótimo de operação, exigindo 4 leituras válidas consecutivas dentro do limiar programado (*thresholdCm*, ajustável de 5 a 60 cm). Isso resulta em um tempo de validação de apenas 180 ms, totalmente imperceptível ao animal, com eliminação de 100% dos falsos disparos durante os ensaios de bancada.

4.3 Validação do Sistema de Visão Noturna Infravermelha (IR 850 nm)

O sistema de iluminação invisível noturna foi projetado para contornar a sensibilidade tátil e visual dos felinos, garantindo o registro de imagem sem gerar clarões ou estresse ao animal.

4.3.1 Desempenho do Circuito de Chaveamento e Pré-aquecimento

O circuito utiliza um conjunto de 6 LEDs infravermelhos de alta intensidade (comprimento de onda de 850 nm) chaveados por um MOSFET de potência N-Channel cuja porta (Gate) é controlada pelo pino PCF_IR (P4 do expansor PCF8574).

Para otimizar o consumo de energia e evitar o aquecimento desnecessário do circuito, o firmware v2.0 implementou uma estratégia preditiva de pré-aquecimento do IR (*irHoldUntil*). Quando o sensor ultrassônico detecta a presença de um corpo em um raio equivalente ao dobro do limiar de disparo ($2 \times \text{thresholdCm}$), o firmware liga preventivamente os LEDs IR 300 ms **antes** da captura da foto (*IR_PREHEAT_MS*).

Essa antecipação permite que os algoritmos de Controle Automático de Exposição (AEC) e Ganho Automático (AGC) do sensor OV2640 ajustem a sensibilidade do DSP da

câmera antes da tomada do quadro definitivo. A Tabela 5.3 compara a qualidade de imagem e o contraste fotográfico obtidos sob diferentes configurações de modo IR no firmware.

Tabela 5.3 – Avaliação da qualidade fotográfica capturada pela OV2640 em ambiente totalmente escuro (<0,5 lux).

Modo IR Configurado (cfg.irMode)	Iluminação Noturna Ativa	Nível de Ruído Digital na Foto (Nível de <i>Gain</i>)	Nitidez dos Contornos do Felino	Qualidade para Identificação Visual
IR_OFF	Desligada	Extremo (Imagens pretas/ilegíveis)	Inexistente	Totalmente Inútil
IR_AUTO (Sem Pré-aquecimento)	Ligada no ato do disparo	Moderado a Alto (Exposição descalibrada)	Regular	Aceitável
IR_AUTO (Com Pré-aquecimento 300ms)	Ligada 300ms antes	Baixo (AGC/AEC calibrados pelo DSP)	Excelente (Nitidez total)	Ideal (Padrão do Protótipo)
IR_ON	Permanentemente ligada	Baixo	Excelente	Excelente (Maior consumo de energia)

Fonte: Elaborado pelo autor (2026).

4.4 Estrutura e Gerenciamento do Banco de Dados e Histórico Local

Para atuar como uma ferramenta auditável de grau clínico e comportamental, o sistema registra as métricas em memória não volátil e no cartão MicroSD montado em barramento SDMMC 1-bit (SD_MMC).

4.4.1 Auditoria via Arquivo CSV e Imagens JPEG

A cada evento de aproximação validado, o firmware cria um arquivo de imagem com nomenclatura temporal estruturada no diretório /fotos/ (exemplo: gato_20260920_221030.jpg) e adiciona uma nova linha de registro textual no arquivo /registros.csv.

A listagem a seguir ilustra a estrutura exata do cabeçalho e dos dados persistidos no cartão MicroSD durante os ensaios do protótipo:

- Snippet de código
- `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

Simultaneamente, a RAM do sistema mantém em estrutura do tipo *ring-buffer* (*struct LogEntry logs[150]*) os 150 registros mais recentes para navegação instantânea pela IHM. O salvamento dos parâmetros de calibração do usuário (*thresholdCm*, *cooldownS*, *irMode*, *ldrInvert*, *confirmReads*) é gerenciado pela biblioteca *Preferences* diretamente na partição *NVS* da memória *Flash*, garantindo retenção integral mesmo após falhas ou desligamento do suprimento de energia.

4.5 Avaliação do Consumo de Energia e Modos de Operação

Sendo o HydraPet Vision um protótipo de bancada offline com foco em alta disponibilidade, a eficiência energética do sistema foi aferida sob tensão de operação contínua de 5,0 V DC fornecida via fonte regulada externa. A Tabela XX detalha a corrente elétrica consumida em cada estado operacional do dispositivo v2.0.

Tabela XX – Perfil de consumo elétrico do protótipo HydraPet Vision v2.0 (Tensão nominal: 5,0 V DC).

Estado de Operação do Protótipo	Periféricos Ativos no Estado	Corrente Média Medida (mA)	Potência Consumida (W)
Repouso Diurno (Monitorando)	ESP32-CAM, Display TFT, HC-SR04, RTC	165 mA	0,825 W
Repouso Noturno (Monitorando)	ESP32-CAM, Display TFT, HC-SR04, LDR	170 mA	0,850 W
Pré-aquecimento Noturno (IR Ativo)	ESP32-CAM, Display TFT, 6 LEDs IR (850nm)	310 mA	1,550 W
Captura e Gravação no SD (Pico)	Câmera OV2640, Gravação SDMMC, LEDs IR	385 mA	1,925 W

Fonte: Elaborado pelo autor (2026).

Os resultados demonstram um consumo elétrico extremamente contido, com pico máximo inferior a 2,0 W no momento exato da gravação fotográfica com iluminação infravermelha ativa. Esse perfil ratifica a viabilidade técnica para que, em evoluções futuras do produto, o equipamento seja alimentado por conjuntos de baterias de lítio (Li-Ion 18650) ou pequenos módulos de recarga solar.

4.6 Levantamento de Custos e Viabilidade Econômica do Protótipo

Um dos pilares norteadores do projeto HydraPet Vision foi o desenvolvimento de uma solução de baixo custo, acessível a tutores e clínicas veterinárias, mantendo elevado padrão de rigor técnico e confiabilidade de dados.

A Tabela 5.5 apresenta a listagem completa, discriminada e itemizada de todos os módulos eletrônicos, componentes passivos, semicondutores e elementos estruturais empregados na fabricação do protótipo v2.0 em valores nominais do mercado nacional.

Tabela 5.5 – Tabela detalhada de custos de materiais e componentes do protótipo HydraPet Vision v2.0.

Item	Componente / Módulo Eletrônico	Especificação Técnica / Detalhes	Qtd.	Custo Unit. Aprox. (R\$)	Custo Total (R\$)
1	Placa AI-Thinker ESP32-CAM	SoC ESP32 Dual-Core + PSRAM 4MB + Conector FPC	1	R\$ 55,00	R\$ 55,00
2	Módulo Câmera OV2640	Resolução 2MP, Lente UXGA/QVGA padrão	1	R\$ 28,00	R\$ 28,00
3	Display TFT ILI9341 2.4"	Resolução 320x240 pixels, Interface SPI, SPI Touch	1	R\$ 48,00	R\$ 48,00
4	Módulo Expansor PCF8574	Expansor de E/S I2C de 8 bits (Endereço 0x20)	1	R\$ 12,00	R\$ 12,00
5	Módulo RTC DS1307	Relógio de Tempo Real I2C + Bateria CR2032	1	R\$ 14,00	R\$ 14,00
6	Sensor Ultrassônico HC-SR04	Faixa de medição 2cm - 400cm, Alimentação 5V	1	R\$ 11,00	R\$ 11,00
7	Módulo Sensor LDR	Fotorresistor com comparador digital LM393	1	R\$ 8,00	R\$ 8,00
8	Cartão MicroSD 16GB Ultra	Classe 10, Formatação FAT32 (SDMMC 1-bit)	1	R\$ 25,00	R\$ 25,00
9	LEDs Infravermelhos 850 nm	LED IR 5mm Alta Potência / Iluminação Noturna	6	R\$ 1,20	R\$ 7,20
10	Transistor MOSFET IRFZ44N / 2N7000	Chaveamento N-Channel para carga dos LEDs IR	1	R\$ 3,50	R\$ 3,50
11	Botões Push-Button Tactil (12x12mm)	Microchaves para navegação IHM (Avança/OK/Volta)	3	R\$ 1,00	R\$ 3,00
12	Resistor 1 kΩ 1/4W	Parte do divisor resistivo do pino ECHO e Gate	2	R\$ 0,20	R\$ 0,40

13	Resistor 2 kΩ 1/4W	Parte do divisor resistivo de adequação de sinal 3.3V	1	R\$ 0,20	R\$ 0,20
14	Resistor 100 Ω 1/4W	Resistor de limitação de corrente de Gate	1	R\$ 0,20	R\$ 0,20
15	Resistor 100 kΩ 1/4W	Resistor de Pull-Down do MOSFET IR	1	R\$ 0,20	R\$ 0,20
16	Resistores de Limitação dos LEDs IR	Limitadores de corrente para arranjo IR (22 Ω a 47 Ω)	3	R\$ 0,20	R\$ 0,60
17	Placa Padrão Perfurada / Protoboard	Placa de fenolite para soldagem do circuito auxiliar	1	R\$ 12,00	R\$ 12,00
18	Barramentos de Pinos / Connectors	Headers fêmea e macho / Cabos Jumpers Dupont	1 Conj.	R\$ 10,00	R\$ 10,00
19	Encapsulamento Impresso 3D (PLA)	Case do protótipo em impressão FDM (Filamento PLA)	150g	R\$ 0,15/g	R\$ 22,50
20	Fonte de Alimentação 5V / 2A Micro-USB	Regulador de parede com alta estabilidade de tensão	1	R\$ 22,00	R\$ 22,00
VALOR TOTAL	Custo Global do Protótipo HydraPet Vision v2.0				R\$ 303,80

Fonte: Elaborado pelo autor (2026).

O custo total de materiais para montagem da Prova de Conceito somou R\$ 303,80, valor expressivamente inferior aos raros dispositivos comerciais de monitoramento de animais de estimação disponíveis no mercado global (cujos preços costumam ultrapassar R\$ 1.500,00 e geralmente não possuem auditoria por imagem nem iluminação infravermelha dedicada). Em uma escala de produção industrial, estima-se que a substituição de módulos avulsos por uma única Placa de Circuito Impresso (PCI) dedicada e o ganho de escala na injeção plástica reduzam o custo unitário de produção para patamares abaixo de R\$ 130,00, confirmando a alta viabilidade comercial e mercadológica da solução.

4.7 Discussão dos Resultados e Perspectivas para Expansão Online

A análise conjunta dos resultados obtidos ao longo do desenvolvimento e dos testes do HydraPet Vision 2.0 permite considerar que a proposta inicial do trabalho foi alcançada. Os resultados demonstram a viabilidade de utilizar uma plataforma embarcada de baixo custo para

realizar o monitoramento e o registro dos eventos relacionados à hidratação felina, mantendo o funcionamento de forma local e sem depender de uma conexão permanente com a internet.

A escolha por uma arquitetura com operação offline mostrou-se particularmente importante para o funcionamento do protótipo. Ao manter o processamento e o armazenamento dos registros no próprio equipamento, o sistema não fica condicionado à disponibilidade da rede Wi-Fi ou ao funcionamento do roteador doméstico. Dessa forma, eventuais oscilações ou interrupções na conexão não interferem diretamente na coleta das informações, permitindo que os eventos continuem sendo registrados localmente.

Essa característica também contribui para o controle sobre os dados gerados pelo sistema. As fotografias capturadas durante o monitoramento permanecem armazenadas no cartão MicroSD instalado no próprio equipamento, ficando sob controle do tutor e não sendo enviadas automaticamente para serviços externos. Além de atender à proposta de funcionamento independente da internet, essa arquitetura reduz a necessidade de transmissão de imagens do ambiente doméstico para servidores externos, o que representa uma consideração relevante em relação à privacidade dos dados.

Outro aspecto observado está relacionado à capacidade de resposta da plataforma. A utilização de interrupções de hardware para a leitura do sinal de eco do HC-SR04, associada à execução periódica das rotinas do firmware, evita que o processamento fique concentrado em operações bloqueantes. Essa organização permite que o sistema continue realizando suas diferentes funções enquanto monitora os sensores, atualiza a interface e executa as demais rotinas previstas no firmware. Como consequência, a interface gráfica apresenta uma resposta mais contínua durante a utilização do equipamento.

A arquitetura adotada, portanto, estabelece um equilíbrio entre autonomia, confiabilidade de coleta e capacidade de resposta. A operação local atende às necessidades atuais do protótipo e, ao mesmo tempo, não impede sua evolução para uma arquitetura conectada. Em uma etapa futura, os dados armazenados localmente poderão ser sincronizados com serviços online por meio da conectividade disponível na ESP32, utilizando protocolos de comunicação adequados à arquitetura IoT.

Essa expansão poderia permitir o acompanhamento remoto dos registros, a visualização das fotografias e a construção de históricos de utilização do bebedouro em uma plataforma web ou aplicativo móvel. A conectividade, entretanto, pode ser incorporada como uma camada adicional ao sistema, mantendo o armazenamento local como mecanismo principal de segurança contra perda de dados. Dessa forma, uma eventual versão conectada poderá

aproveitar a infraestrutura desenvolvida na versão 2.0 sem comprometer a autonomia que caracteriza o funcionamento atual do protótipo.

4.7.1 Arquitetura Preparada para Atualização Online Futurível

Apesar da operação atual do protótipo ser totalmente independente de conexões de rede, o desenvolvimento do hardware e do firmware foi projetado visando à facilidade de migração para um modelo híbrido ou totalmente online (*cloud-connected*).

A presença nativa do hardware Wi-Fi/Bluetooth no SoC ESP32 possibilita que futuras revisões do firmware adicionem camadas de software sem a necessidade de alteração de esquemático elétrico ou troca de componentes. A Tabela 5.6 correlaciona os recursos atualmente implementados no protótipo v2.0 às expansões online planejadas para a versão 3.0.

Tabela 5.6 – Roteiro de expansão funcional: Comparativo entre o protótipo offline v2.0 e as atualizações online planejadas para a versão v3.0.

Módulo / Funcionalidade	Implementação Atual no Protótipo v2.0 (Offline)	Expansão Projetada para Atualização v3.0 (Online)
Banco de Dados	Armazenamento local CSV no MicroSD e NVS Flash	Sincronização automática com banco de dados em Nuvem (Firebase / AWS IoT)
Comunicação de Dados	Leitura local de logs pela tela TFT	Telemetria em tempo real via protocolo MQTT / HTTP REST API
Notificação de Alertas	Indicadores visuais na IHM (Status/Pill no display)	Notificações <i>Push</i> no smartphone (WhatsApp Bot / App Mobile) em caso de baixa hidratação
Ajuste de Relógio	Configuração manual de Data/Hora pelo Menu TFT	Sincronização automática de horário via servidor NTP via Wi-Fi
Visualização de Fotos	Galeria local decodificada no display TFT 2.4"	Transmissão de imagens via <i>Web Server</i> embarcado ou armazenamento <i>S3 Cloud</i>
Inteligência Artificial	Deteção física de proximidade (HC-SR04)	Classificação facial/recompensa do felino por rede neural (TinyML / ESP-WHO)

Fonte: Elaborado pelo autor (2026).

Essa modularidade assegura que o projeto HydraPet Vision atinja com pleno êxito o seu papel acadêmico e de engenharia: consolidar-se como uma plataforma robusta, testada e auditável de monitoramento da saúde felina, pronta para transitar de uma Prova de Conceito offline de bancada para um produto comercial IoT de alto impacto no mercado *pet tech*.

4.8 Síntese dos Resultados e Considerações do Capítulo

A avaliação experimental e o conjunto de ensaios de engenharia realizados sobre o protótipo HydraPet Vision v2.0 comprovaram a eficiência e a viabilidade da arquitetura proposta. A integração do microcontrolador ESP32-CAM com periféricos dedicados via barramento I2C compartilhado (SCCB/Wire) superou as limitações técnicas de E/S observadas na versão inicial do projeto. A utilização do expensor de portas PCF8574, somada à alocação estratégica do display TFT ILI9341 no barramento HSPI a 26 MHz, permitiu a construção de uma IHM completa, reativa e sem conflitos de hardware.

Sob a ótica do sensoriamento, a substituição de chamadas bloqueantes pela técnica de amostragem temporal por interrupção de hardware no pino ECHO (*echoISR*), combinada ao algoritmo de validação por quatro leituras consecutivas (*confirmReads* = 4), eliminou integralmente os falsos disparos decorrentes de ruídos acústicos e ambientais. O tempo total de validação de 180 ms garantiu a captura da presença do felino sem causar estranhamento ou atraso perceptível no acionamento do sistema.

No âmbito da aquisição visual, o circuito de iluminação infravermelha noturna (850 nm) com pré-aquecimento preditivo de 300 ms (*IR_PREHEAT_MS*) otimizou a calibração prévia dos algoritmos internos de controle de ganho e exposição (AGC/AEC) do sensor OV2640. O resultado traduziu-se em registros fotográficos de alta nitidez e contraste em ambientes de escuridão total ($< 0,5 \text{ lux}$), preservando a saúde visual do animal.

A arquitetura de armazenamento baseada em duplo registro — retenção local em arquivos estruturados CSV e imagens JPEG no cartão MicroSD (SDMMC 1-bit) combinada ao *ring-buffer* na memória RAM para consulta rápida via IHM — estabeleceu um mecanismo auditável e confiável de dados comportamentais. Essa estrutura manteve-se íntegra mesmo sob cenários simulados de falha de alimentação elétrica, demonstrando a robustez da operação *offline*.

Do ponto de vista econômico e energético, o dispositivo apresentou um consumo de pico contido (385 mA / 1,925 W durante o disparo com IR ativo) e um custo total de fabricação de R\$ 303,80 para o modelo de prova de conceito. Esse valor reafirma a acessibilidade da solução em comparação às alternativas de mercado e sinaliza forte viabilidade para produção em escala.

Por fim, embora o protótipo tenha sido intencionalmente consolidado como uma solução autônoma *offline* para assegurar imunidade a falhas de rede, a estrutura modular do firmware e a presença nativa do rádio Wi-Fi/Bluetooth no SoC ESP32 deixam o sistema plenamente preparado para expansões futuras. A inclusão de protocolos de telemetria em nuvem

(MQTT/HTTP) e a integração com aplicações móveis foram devidamente mapeadas como o próximo passo natural de evolução do produto.

Com os resultados operacionais, quantitativos e econômicos devidamente validados e discutidos neste capítulo, o trabalho avança para o Capítulo 6, no qual serão apresentadas as conclusões finais do projeto, o cumprimento dos objetivos estabelecidos e as propostas de trabalhos futuros.

5. CONCLUSÃO

O desenvolvimento, a implementação prática e a validação do protótipo HydraPet Vision, em sua versão 2.0, demonstraram a viabilidade técnica de uma solução autônoma e de baixo custo voltada ao monitoramento da hidratação felina. 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 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 passou a funcionar como uma estação local de monitoramento, capaz de realizar a detecção da aproximação do animal, confirmar o evento por meio de leituras consecutivas, controlar a iluminação infravermelha, realizar o registro fotográfico, armazenar as imagens e manter um histórico dos eventos. Essa integração entre sensores, câmera, armazenamento e interface local permitiu transformar o protótipo experimental em um sistema mais organizado e adequado ao uso contínuo no ambiente doméstico.

5.1 Síntese das conquistas e resultados da versão 2.0

Uma das principais soluções adotadas na versão 2.0 foi a utilização do expensor de portas digitais PCF8574 no barramento I2C. Essa implementação permitiu ampliar a quantidade de entradas e saídas disponíveis sem comprometer as conexões utilizadas pela câmera, pelo display TFT e pelo cartão MicroSD. Por meio do expensor, foram integrados os três botões da interface, o acionamento do sensor ultrassônico, a leitura do módulo LDR e o controle do circuito de iluminação infravermelha. O RTC DS1307 também permanece integrado ao mesmo

barramento I2C, formando uma arquitetura que aproveita melhor 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, 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. Para aumentar a confiabilidade da detecção, também foi incorporada a confirmação por leituras consecutivas, reduzindo a ocorrência de disparos provocados por ruídos acústicos ou reflexões ocasionais do sinal.

O sistema de iluminação infravermelha também foi integrado ao processo de captura das imagens. A utilização de seis LEDs de 850 nm, acionados por meio de um transistor MOSFET, possibilita a realização 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 sem utilizar iluminação branca convencional. O controle da iluminação é realizado de acordo com as condições identificadas pelo sensor LDR e pelas configurações selecionadas no sistema, permitindo o funcionamento automático ou manual conforme a necessidade.

A interface homem-máquina (IHM) desenvolvida com o display TFT ILI9341 de 2,4 polegadas também contribuiu para tornar o equipamento mais independente. Por meio dela, o usuário consegue 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. Dessa forma, não é necessário utilizar um computador para realizar as principais operações de configuração e acompanhamento do protótipo.

A persistência dos dados foi outro aspecto importante da versão 2.0. As configurações do sistema são armazenadas na memória não volátil por meio da biblioteca *Preferences*, permitindo que os parâmetros definidos pelo usuário sejam recuperados após o desligamento ou uma interrupção da alimentação. Os registros dos eventos são armazenados no arquivo *registros.csv*, enquanto as fotografias são organizadas no diretório /fotos do cartão MicroSD. Essa estrutura permite relacionar cada evento registrado com sua respectiva imagem, data, horário e distância medida, proporcionando maior organização e rastreabilidade dos dados coletados.

O desenvolvimento do gabinete por meio de impressão 3D também contribuiu para a evolução do protótipo. A fabricação pelo processo FDM permitiu acomodar os componentes eletrônicos em uma estrutura única, protegendo a montagem contra respingos e facilitando sua

utilização no ambiente doméstico. Além da proteção dos componentes, a possibilidade de modificar o modelo tridimensional facilita futuras alterações no projeto, caso sejam incorporados novos sensores, módulos ou recursos ao sistema.

De maneira geral, essas modificações permitiram que a versão 2.0 apresentasse uma arquitetura mais integrada e funcional em relação à proposta inicial. A combinação entre sensoriamento, confirmação do evento, iluminação, captura fotográfica, armazenamento local e interface gráfica estabelece uma base tecnológica adequada para o acompanhamento automatizado dos eventos de hidratação felina.

5.2 Trabalhos e evoluções futuras

Embora a versão 2.0 tenha alcançado uma estrutura funcional e estável para o monitoramento local, o projeto foi concebido de maneira modular, permitindo a incorporação de novos recursos em versões futuras. As possibilidades de evolução concentram-se principalmente na identificação individual dos animais, na obtenção de medidas quantitativas de consumo, no aumento da autonomia energética e na disponibilização dos dados por meio de serviços em nuvem e aplicativos móveis.

Uma das possibilidades de maior interesse é a incorporação de técnicas de Visão Computacional e Aprendizado de Máquina diretamente no dispositivo, utilizando conceitos de Edge AI. Essa evolução seria especialmente relevante em residências onde há mais de um gato, pois os registros atualmente obtidos não permitem identificar automaticamente qual animal realizou determinado acesso ao recipiente de água.

Para essa finalidade, poderiam ser utilizadas Redes Neurais Convolucionais (CNNs) otimizadas e quantizadas, com suporte de tecnologias como o TensorFlow Lite for Microcontrollers. A partir de uma base de imagens dos animais, considerando características faciais e padrões de pelagem, seria possível desenvolver um mecanismo de identificação individual. Com isso, os registros poderiam ser associados a cada animal, permitindo a construção de históricos separados de utilização do reservatório de água.

Outra possibilidade de evolução está relacionada à medição efetiva da quantidade de água consumida. Para isso, poderia ser incorporada uma célula de carga associada ao conversor e amplificador HX711, instalada na base do reservatório. Essa configuração permitiria acompanhar a variação de massa do recipiente ao longo do tempo e relacioná-la aos eventos registrados pelo sistema. Dessa forma, além de contabilizar os acessos e realizar o registro

fotográfico, o protótipo poderia obter uma estimativa mais precisa da quantidade de água retirada do reservatório.

Também poderia ser desenvolvido um sistema de alimentação de emergência utilizando células de íon-lítio, como a 18650, associado a um sistema de gerenciamento de bateria (BMS). Essa implementação teria como objetivo manter o monitoramento durante interrupções ou oscilações na alimentação elétrica residencial, aumentando a disponibilidade do equipamento e reduzindo a possibilidade de perda de registros.

Em uma etapa posterior de desenvolvimento, a estrutura produzida por impressão 3D poderia ser substituída por um gabinete produzido por injeção plástica. A utilização de materiais adequados ao ambiente de aplicação, incluindo polímeros com resistência à umidade e proteção contra radiação ultravioleta, poderia proporcionar maior padronização do produto e facilitar uma eventual produção em maior escala. Essa possibilidade, entretanto, dependeria de novos estudos de engenharia de produto, custos, materiais, vedação e processos de fabricação.

Outra linha de evolução envolve a conectividade. Embora a versão 2.0 tenha sido desenvolvida para funcionar localmente *offline* e sem depender de uma conexão permanente com a internet, os recursos de comunicação disponíveis na plataforma ESP32 permitem ampliar o sistema para uma arquitetura IoT conectada. Os registros poderiam ser enviados de forma assíncrona por Wi-Fi ou Bluetooth Low Energy utilizando protocolos como MQTT ou HTTP/REST e, posteriormente, armazenados em plataformas de IoT, como AWS IoT, TagoIO ou ThingsBoard.

A disponibilização dessas informações em um aplicativo móvel também representaria uma evolução importante. Um aplicativo para Android e iOS poderia apresentar o histórico dos eventos de forma gráfica, permitindo ao tutor acompanhar os registros ao longo dos dias e consultar as fotografias associadas a cada ocorrência. A aplicação também poderia utilizar notificações para informar determinados eventos, desde que os critérios de funcionamento fossem previamente definidos.

Com uma quantidade maior de dados históricos, outra possibilidade seria utilizar técnicas de análise de dados e modelagem preditiva para identificar alterações no padrão de utilização do reservatório de água. Variações relevantes na frequência dos acessos ou no comportamento registrado poderiam gerar alertas para o tutor. Esses alertas deveriam ser entendidos como mecanismos de acompanhamento e apoio, e não como diagnóstico veterinário, sendo necessária a avaliação de um profissional habilitado diante de alterações persistentes no comportamento do animal.

5.3 Considerações finais

A trajetória de desenvolvimento do HydraPet Vision 2.0 demonstra como os conhecimentos de sistemas embarcados podem ser aplicados à criação de soluções voltadas ao monitoramento e ao bem-estar animal. O projeto enfrentou limitações importantes relacionadas à disponibilidade de GPIOs, à integração de diferentes periféricos, ao sensoriamento e ao armazenamento de dados, resultando em uma arquitetura capaz de reunir essas funções em um único protótipo.

A principal contribuição do trabalho está na integração entre a detecção automática da aproximação do animal, a confirmação do evento, o registro fotográfico, o armazenamento das informações e a disponibilização dos dados por meio de uma interface local. Essa abordagem permite substituir parte do acompanhamento manual por um processo automatizado de coleta e organização das informações, criando um histórico que pode ser consultado posteriormente.

Os resultados obtidos também estabelecem uma base para novas pesquisas e aprimoramentos. A incorporação de identificação individual por visão computacional, medição volumétrica, conectividade em nuvem, aplicativos móveis e técnicas de análise de dados poderá ampliar significativamente as possibilidades do sistema. Essas evoluções poderão transformar o protótipo em uma plataforma mais completa de acompanhamento do comportamento hídrico felino, desde que sejam submetidas a novas etapas de desenvolvimento, testes e validação.

Dessa forma, o HydraPet Vision 2.0 representa não apenas o resultado da implementação de um sistema embarcado, mas também uma base tecnológica para estudos futuros que integrem Internet das Coisas, visão computacional e monitoramento animal. O trabalho demonstra a possibilidade de utilizar recursos eletrônicos de baixo custo para coletar e organizar informações que podem contribuir para o acompanhamento da rotina dos animais e servir de apoio a futuras aplicações na área de tecnologia aplicada à medicina veterinária.

REFERÊNCIAS BIBLIOGRÁFICAS

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

ESPRESSIF SYSTEMS. **ESP32 Series Datasheet**. V3.9. 2023. Disponível em: <https://www.espressif.com/>. Acesso em: [Data de Acesso].

ADAFRUIT INDUSTRIES. Arduino: Adafruit ILI9341 library. Adafruit Learning System, [s.d.].

ANALOG DEVICES. DS1307: 64 x 8, serial, I²C real-time clock. [S.l.]: Analog Devices, 2015.

ARDUINO. Language Reference. [S.l.]: Arduino, [s.d.].

BAHGA, Arshdeep; MADISETTI, Vijay. Internet of Things: a hands-on approach. [S.l.]: VPT, 2014.

BROWN, C. A.; ELLIOTT, J.; SCHMIEDT, C. W.; BROWN, S. A. Chronic kidney disease in aged cats: clinical features, morphology, and proposed pathogeneses. *Veterinary Pathology*, v. 53, n. 2, p. 309-326, 2016. DOI: 10.1177/0300985815622975.

BURGESS, Phillip. Adafruit GFX graphics library. Adafruit Learning System, 2012.

DEITEL, Harvey M. C++: como programar. 5. ed. São Paulo: Pearson Universidades, 2006.

DI BARTOLA, Stephen P. Fluid, electrolyte, and acid-base disorders in small animal practice. 4. ed. St. Louis: Elsevier Saunders, 2012.

ELECFREAKS. HC-SR04 ultrasonic module user guide. [S.l.]: Elecfreaks, [s.d.].

ESPRESSIF SYSTEMS. ESP32 series datasheet. [S.l.]: Espressif Systems, 2026.

GRANT, David C. Effect of water source on intake and urine concentration in healthy cats. *Journal of Feline Medicine and Surgery*, v. 12, n. 6, p. 431-434, 2010. DOI: 10.1016/j.jfms.2009.10.008.

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

HAWTHORNE, A. J.; MARKWELL, P. J. Dietary sodium promotes increased water intake and urine volume in cats. *Journal of Nutrition*, v. 134, n. 8, p. 2128S-2129S, 2004. DOI: 10.1093/jn/134.8.2128S.

KOPECNY, L. et al. Urolithiasis in cats: evaluation of trends in urolith composition and risk factors from 2005 to 2018. *Journal of Veterinary Internal Medicine*, v. 35, p. 1397-1405, 2021. DOI: 10.1111/jvim.16121.

KOSMAL, Pauline A. L.; SHOVELLER, Anna K.; GILLIES, Genevieve; ARMSTRONG, Lawrence E. Diet format, protein, amino acids, salt, and osmolytes, as well as water viscosity, affect water consumption in domestic cats: a scoping review of 32 publications (published from 1975 to 2025) on water intake, hydration status, and related health outcomes. *Journal of Animal Science*, v. 104, 2026. DOI: 10.1093/jas/skaf434.

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

LIMA, Charles Borges de; VILLAÇA, Marco V. M. AVR e Arduino: técnicas de projeto. São Paulo: Clube de Autores, 2012.

LIU, Jane W. S. Real-time systems. Upper Saddle River: Prentice Hall, 2000.

LITTLE, Susan E. (ed.). The cat: clinical medicine and management. St. Louis: Elsevier, 2011.

MONK, Simon. Programação com Arduino: começando com sketches. Porto Alegre: Bookman, [s.d.].

MONK, Simon. Programação com Arduino II: passos avançados com sketches. Porto Alegre: Bookman, 2015.

OLIVEIRA, André Schneider de; ANDRADE, Fernando Souza de. Sistemas embarcados: hardware e firmware na prática. 2. ed. São Paulo: Érica, 2012.

REIS, Paulo J. et al. How cats lap: water uptake by *Felis catus*. *Science*, v. 330, n. 6008, p. 1231-1234, 2010. DOI: 10.1126/science.1195421.

ROBBINS, Michael T.; CLINE, Martha G.; BARTGES, Joseph W.; FELTY, Erin; SAKER, Korinn E.; BASTIAN, Richard; WITZEL, Angela L. 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, Andrew H.; CANEY, Sarah; CHALHOUB, Serge; ELLIOTT, Jonathan; FINCH, Natalie; GAJANAYAKE, Isuru; LANGSTON, Catherine; LEFEBVRE, Hervé P.; WHITE, Joanna; QUIMBY, Jessica. 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.

TAYLOR, Samantha; FINCH, Natalie; CANEY, Sarah; ELLIOTT, Jonathan; GEDDES, Rebecca; MORTIER, Femke; PARKER, Valerie; QUIMBY, Jessica; SEGEV, Gilad; WHITE, Joanna. 2026 iCatCare consensus guidelines on the diagnosis and management of chronic kidney disease in cats. *Journal of Feline Medicine and Surgery*, v. 28, n. 9, 2026. DOI: 10.1177/1098612X261466553.

TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L. Sistemas digitais: princípios e aplicações. 11. ed. São Paulo: Pearson Prentice Hall, 2011. WHITE, Joanna D.; MALIK, Richard; NORRIS, Jacqueline M. Feline chronic kidney disease: can we move from treatment to prevention? The Veterinary Journal, v. 190, n. 3, p. 317-322, 2011. DOI: 10.1016/j.tvjl.2010.12.011.

ZANGHI, Brian M.; GERHEART, Lynn; GARDNER, Cari 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. DOI: 10.2460/ajvr.79.7.733.

ZANGHI, Brian M.; WILS-PLOTZ, Emma; DEGEER, Staci L.; GARDNER, Cari L. Effects of a nutrient-enriched water with and without poultry flavoring on water intake, urine specific gravity, and urine output in healthy domestic cats fed a dry kibble diet. American Journal of Veterinary Research, v. 79, n. 11, p. 1150-1159, 2018. DOI: 10.2460/ajvr.79.11.1150.