

Monitoramento Inteligente de ambientes para Animais Domésticos

Gabriela Goes do Nascimento -180120867

Faculdade do Gama
St. Leste Projeção A - Gama Leste
Brasília - DF, 72444-240
Email: 180180867@aluno.unb.br

Pamela Raquel Almeida Nogueira Campagnucci -
180026097

Faculdade do Gama
St. Leste Projeção A - Gama Leste
Brasília - DF, 72444-240
Email: 180026097@aluno.unb.br

I. RESUMO

Este trabalho desenvolve um monitoramento autônomo de ambientes para animais domésticos, usando Raspberry Pi como base de hardware e Raspberry Pi OS como base de software. Códigos e funcionalidades são produzidos em *Shell Script* e em linguagem C. O objetivo principal é criar ferramentas de verificação do estado do ambiente onde encontra-se o animal doméstico e enviá-lo ao seu dono, sanando uma demanda do mercado. O uso da ferramenta git como auxiliar ao desenvolvimento visa familiarizar o aluno com esta opção de gestão de projetos e versões, bem como cuidar para que os códigos sejam salvos a cada etapa, evitando perda de dados.

II. INTRODUÇÃO

Em 31 de dezembro de 2019, a Organização Mundial da Saúde (OMS) foi informada sobre uma nova pandemia mundial que estaria se espalhando por todos os países do mundo, o novo coronavírus (OPAS, 2019). Esse vírus se espalha através do contato com as secreções de um paciente infectado, por conta disso as pessoas foram forçadas a ficar em quarentena até que uma vacina ou medicamento fosse colocado no mercado. Isso ocasionou um aumento expressivo no número de animais de estimação adotados, a União Internacional Protetora dos Animais (UIPA), apresentou uma pesquisa em que a procura por adoção de animais aumentou 400% durante o primeiro trimestre de 2020.

Porém após a população ser vacinada e as taxas de contágio e de óbitos caírem, a população voltou a trabalhar e passar menos tempo em suas casas, por conta disso os animais de estimação acabam ficando estressados, sendo importante acompanhar como o animal está se comportando fora da presença do dono.

Alguns aplicativos recentes tem a proposta de os donos pagarem para alguém sair com seu animal, passear, brincar e

até mesmo se alimentar. Porém o dono não tem como saber se o animal está sendo tratado corretamente e se o ambiente está adequado ao animal.

Outro problema encontrado é em relação ao banho e tosa dos animais de estimação ou tratamentos veterinários, onde o animal é submetido a um ambiente onde o dono não tem acesso e muitas vezes é estressante. Há relatos de clientes que tiveram seus pets machucados no processo de banho e tosa, e mesmo reclamando aos donos do estabelecimento, eles não quiseram se responsabilizar pelos danos, já que não tem provas do ocorrido.

Para solucionar esse problema, é proposto um sistema inteligente de medição, que avisará se o animal está em um ambiente adequado e deixará ciente ao dono qual situação atual do animal.

O modelo proposto atende a uma demanda ainda não sanada de mercado e tendo em vista que o número de animais domésticos no Brasil é o quarto maior do mundo – 132 milhões; 2º em número de cães, gatos e aves canoras e ornamentais; 9º em número de répteis e pequenos mamíferos; 10º em número de peixes ornamentais (IBGE, 2013) e os gastos com animais domésticos cresceram 129% no ano de 2021 (*Domestic View*, 2022). Isto posto, a proteção destes durante idas ao banho e tosa, hotéis para cães, ou quando estão sozinhos em casa, torna-se extremamente necessária.

Ademais, o sistema permite trazer mais segurança, pois o animal doméstico é monitorado a todo instante, e com isto, o dono pode intervir, caso algo grave aconteça, pois ele pode verificar se a temperatura, umidade do ar e oxigênio estão em níveis adequados para o seu *pet*.

A Intelbras desenvolveu uma câmera que monitora o pet em tempo real, com o dono tendo acesso ao vídeo do seu pet através do celular. “Uma das maiores preocupações de quem é tutor de um animal doméstico é deixá-lo sozinho em casa. É aí que entra a câmera para vigiar pet, uma solução inovadora para manter seu bichinho seguro mesmo de longe.” (INTELBRAS, 2022).

A empresa *SurePetCare* lançou um dispositivo que você adiciona à coleira do pet para monitorar atividades e comportamentos do animal. “O Monitor de Comportamento e Atividades com App Sure PetCare Animo para Cachorros aprende e interpreta a atividade única e os padrões de comportamento do seu cão. Está com eles 24 horas por dia, dando-lhe uma maior compreensão da sua vida. Caminhadas matinais, dormindo embaixo da mesa, latindo para o carteiro. Os dias do seu cão são cheios de ação e agora você fica sabendo de tudo.” (SUREPETCARE).

Este projeto objetiva desenvolver um monitoramento inteligente de ambientes para animais domésticos, em que o usuário possa receber dados do sistema embarcado com os sensores e que tenha a possibilidade de ser adaptado e utilizado em qualquer local em que estes animais estejam.

O sistema deve medir a temperatura e a umidade do ar e o nível de oxigênio, tirar fotos de cinco em cinco minutos do animal e enviar estes dados ao seu dono.

III. DESENVOLVIMENTO

A. Hardware

Para o funcionamento do sistema será necessário uma raspberry pi, para controlar o envio e recebimento dos parâmetros, sensor de temperatura e umidade, sensor de gases nocivos e uma câmera, para monitorar o estado do animal a ser transportado.

- 1) *Raspberry Pi*: Para este projeto é utilizada a Raspberry Pi 3 Modelo B que traz várias melhorias diante das versões anteriores. A placa conta com processador Broadcom Quad Core BCM2837 de 64 bits, porém com um encapsulamento que o torna mais eficiente em dissipação de calor, isso faz com que o processador agora seja capaz de operar em 1.4GHz[3].

Será utilizada deste microcontrolador, a CPU, o armazenamento para arquivos e programas, internet - com ou sem fio, a entrada usb, para a comunicação com a *webcam* para tirar as fotos, dois GPIO para comunicação digital com os sensores e os pinos VCC e GND

Para instalar o sistema operacional foi utilizado um cartão SD de 64 Gb de memória.

- 2) *DHT11*: O sensor DHT11 foi escolhido para medir a temperatura e umidade do ar por atender os requisitos do projeto, permite fazer leituras de temperaturas entre 0 a 50 Celsius e umidade entre 20 a 90%.
- 3) *MQ135*: Esse sensor é responsável por medir gases nocivos como: amônia, dióxido de carbono, benzeno, óxido nítrico e também fumaça e álcool. O sensor é capaz de detectar concentrações entre 10 e 300 ppm (partes por milhão) de amônia e álcool e entre 10 e

1000 ppm de benzeno no ar.

- 4) *Webcam*: Para fazer a captura das imagens foi utilizada a *Webcam* Logitech C920 Full HD 1080p, a conexão é feita via USB, possui lentes de vidro com cinco elementos, definição Full HD e autofocus proporcionam imagens nítidas, com cores precisas para gravações de até 1080p a 30 quadros por segundo.
- 5) *Alarme sonoro*: Caso os sensores dêem algum alarme informando que o animal corre algum perigo (alerta de gases tóxicos), será disparado um alarme sonoro, para alertar quem estiver próximo ao animal. O alarme sonoro será obtido através de um buzzer simples.

Na Fig. 1, encontra-se o esquemático com as ligações do hardware do projeto.

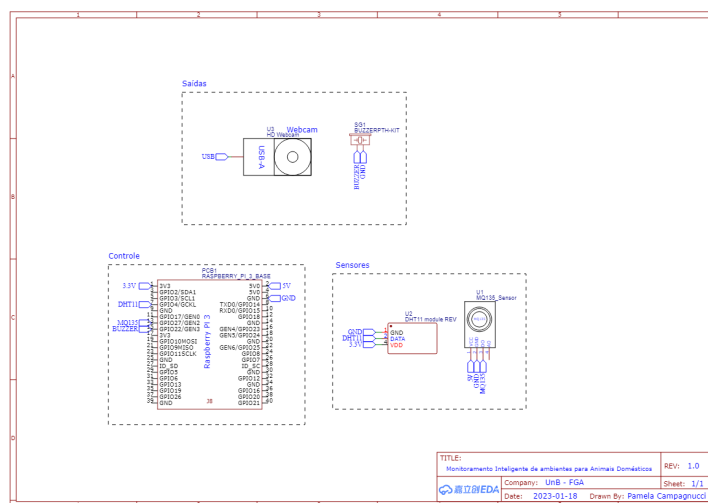


Figura 1: Esquemático. Fonte: Autor. Link: https://drive.google.com/file/d/1blmTGYg0HCVQX8VeIYutCQwXpLEuBnM/view?usp=share_link

Na Fig. 2, é possível visualizar a tabela com o *Bill Of Materials* (BOM).

ID	Nome	Designador	Q	Preço
1	RASPBERRY 3B	PCB1	1	R\$ 565,11
2	BUZZER	SG1	1	R\$ 2,76
3	MQ135	U1	1	R\$ 19,90
4	DHT11 MODULO	U2	1	R\$ 10,92
5	HD Webcam	U3	1	R\$ 269,90

Figura 2: Bill Of Materials. Fonte: Autor

Na Fig. 3, é possível visualizar o diagrama de blocos do hardware.

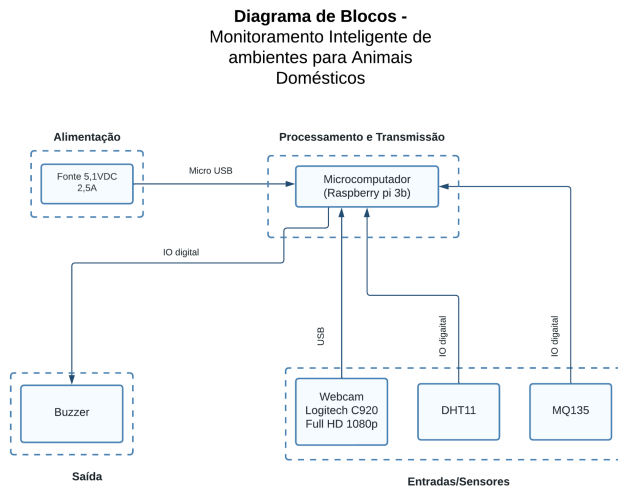


Figura 3: Diagrama de Blocos do Hardware. Fonte: Autor
Link:

https://drive.google.com/file/d/1OupFIMtIBgS60IRygd9_ppk_k6bO4xu0/view?usp=share_link

B. Software

A parte de software do projeto será escrita na linguagem C e em *shell script*. O programa já começará a ser executado após a energização da raspberry pi, sem necessitar de uma interface homem-máquina. Dentro do programa serão feitas as leituras dos sensores e alertas serão habilitados caso as leituras excedam os parâmetros previamente estabelecidos. Quando solicitado pelo usuário, será enviada uma foto e os dados de temperatura e umidade do ar para o dono do animal e caso o sensor de gases detecte algo nocivo, será enviado um alerta pelo *telegram*.

Para analisar se o funcionamento do produto é viável, foram utilizados programas em Python, C e *shell script*, porém o projeto final irá ser desenvolvido somente em C e *shell*, pois utilizando essas linguagens o tempo de resposta do programa é bem menor, proporcionando um projeto mais otimizado e também consumindo menos memória da raspberry pi.

- 1) *Sistema operacional*: O sistema operacional utilizado é o Raspberry Pi OS (antigo Raspbian Desktop), com um ambiente gráfico que facilita a interação do iniciante. Esta é uma variante Linux gratuita baseada no Debian, fruto de um projeto comunitário em constante desenvolvimento com foco em estabilidade e desempenho. É otimizado para execução no Raspberry Pi e pode ser baixado diretamente no site da Fundação Raspberry. Em caso de desenvolvimentos futuros baseados nesta solução, recomenda-se o uso da versão do sistema operacional

sem o ambiente gráfico, visto que este consome recursos desnecessários do hardware. Também para um sistema totalmente dedicado, muitos pacotes pré-instalados podem ser removidos, ou pode-se adotar a versão mínima e instalar apenas os programas que executam as funções e suas dependências.

- 2) *Captura de Imagem*: Para fazer a captura da imagem foi instalado o software *fswebcam*, e foi criado um *script* para fazer a captura automática, e salvar o arquivo na área de trabalho da Raspberry Pi.
- 3) *Envio de Dados*: Optou-se por fazer o envio dos dados ao cliente pelo *Telegram* pois é um *software* de fácil acesso para o consumidor e também por conta do envio da imagem não perder resolução em comparação com outros aplicativos de mensagem como o *Whatsapp*. O código é feito em *shell script* para o envio das imagens e informações, utilizando o protocolo *HTTP* e o comando "*curl*". Quando o usuário clicar em "*info*" no *chat* com o *bot* "*@PetCaregiver*", irá receber os dados com a imagem e a temperatura e umidade do ar.
- 4) *Medição da Temperatura e umidade*: Para ler o sensor de temperatura, foi utilizado um programa em Python, onde ele faz a leitura em graus °C e também lê a umidade, quando solicitado pelo usuário, porém este não faz a leitura de decimais, apenas de números inteiros.
- 5) *Medição de gases nocivos*: Para ler o sensor de gases nocivos será utilizado sua entrada digital, que tem como saída *HIGH* ou *LOW*. Se a leitura for em nível lógico baixo, quer dizer que o sensor está detectando gases nocivos no ambiente, o que será notificado ao responsável do pet por meio do *Telegram*, para modificar o nível de sensibilidade da saída digital, foi ajustado o *trimpot* que fica localizado na parte posterior do sensor.
- 6) *Alarme Sonoro*: Para acionar o alarme sonoro foi utilizado um programa em C, em que o buzzer irá tocar sempre que o sensor de gases nocivos mudarem para nível lógico baixo. Quando o ar estiver sem perigo, o sensor para de tocar.

Na Fig. 4, é possível observar o fluxograma com o algoritmo para a solução do projeto.

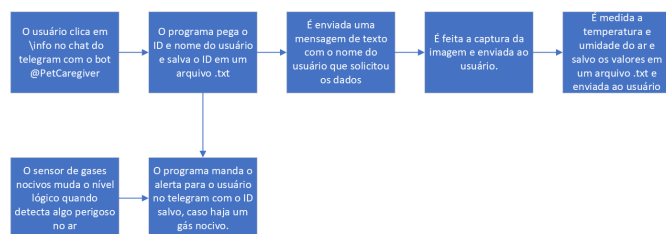


Figura 4: Fluxograma. Fonte: Autor. Link: <https://github.com/gabrielaggn/Projeto-SOE/blob/main/documenta%C3%A7%C3%A3o/Algoritimo%20da%20solu%C3%A7%C3%A3o.png>

IV. RESULTADOS

Ao longo do prazo de projeto, todas as etapas foram devidamente cumpridas e a complexidade necessária ao desenvolvimento foi alcançada. O produto final contempla os vários aspectos necessários à automação das tarefas relacionadas ao cuidado com o animal de estimação que o dono deseja ter ao deixá-lo em algum lugar longe da sua supervisão.

Para validação do projeto, foram feitos testes com os sensores e códigos de envio de forma separada, depois de observar que os códigos funcionam com bibliotecas e algoritmos prontos, foram criados novos códigos para otimizar o projeto, juntá-los para se adaptarem à finalidade almejada.

Após unir os códigos, para automação, estes foram inseridos no arquivo "rc.local" e no "autostart" da pasta ~/config/lxsession/LXDE-pi, para rodá-los na inicialização, depois disso, foi verificado os requisitos.

São considerados satisfatórios os requisitos de:

- Medir temperatura e umidade do ar;
- Medir o nível de gases nocivos do ar;
- Fazer a captura da imagem quando solicitada pelo usuário;
- Enviar um alerta ao usuário quando for detectado algum gás nocivo; e
- Enviar as imagens e dados de temperatura e umidade do ar quando solicitado pelo usuário.

V. CONCLUSÃO

Propor uma solução tecnológica que auxilie os donos de animais domésticos a terem informações sobre eles é uma tarefa primordial para trazer mais segurança a estes animais, especialmente pelo fato do número de pessoas que adquirem *pets* aumentar cada vez mais ao longo dos anos, e com o auxílio do monitoramento inteligente, esta demanda pode ser sanada.

A partir dos resultados e testes executados, observou-se que o projeto cumpriu com os objetivos esperados, conseguindo abranger todos os requisitos.

Quanto aos aspectos teóricos a serem compreendidos e desenvolvidos na disciplina, considera-se que também faz-se suficiente, por ter sido concebido com base no padrão de hardware e de linguagem de programação solicitados. Ainda, por apresentar solução viável e possível ser executada no tempo disponível.

REFERÊNCIAS

- [1] G1RIO. Cachorro morre após voo Rio-SP, e dona culpa a companhia aérea. Globo, [S. l.], p. 1, 21 set. 2021. Disponível em: <https://g1.globo.com/rj/rio-de-janeiro/noticia/2021/09/21/cachorro-morre-apos-voo-rio-sp-e-dona-culpa-a-companhia-aerea-gh.html>. Acesso em: 9 nov. 2022.
- [2] MAGESKY, Lais. Cachorro morre de calor em voo de São Paulo para Vitória. A gazeta, [S. l.], p. 1, 6 ago. 2022. Disponível em: <https://www.agazeta.com.br/es/cotidiano/cachorro-morre-de-calor-em-voo-de-sao-paulo-para-vitoria-0820>. Acesso em: 9 nov. 2022.
- [3] ABINPET. - População de animais de estimação no Brasil - 2013 - Em milhões, [S. l.], 2014. Disponível em: <https://www.gov.br/agricultura/pt-br/assuntos/camaras-setoriais-tematicas/documentos/camaras-tematicas/insumos-agropecuarios/anos-anteriores/ibge-populacao-de-animais-de-estimacao-no-brasil-2013-abinpet-79.pdf>. Acesso em: 10 nov. 2022.
- [4] LONGUINHO, Daniela. Gastos com pets cresceram 129% no 2º ano da pandemia, diz pesquisa. Agência Brasil EBC, [S. l.], p. 1, 16 mar. 2022. Disponível em: <https://agenciabrasil.ebc.com.br/radioagencia-nacional/geral/audio/2022-03/gastos-com-pets-cresceram-129-no-2o-ano-da-pandemia-diz-pesquisa>. Acesso em: 10 nov. 2022.
- [5] GUSE, Rosana. Como funciona o sensor de gás MQ-135?. Brasil, 9 fev. 2022. Disponível em: <https://www.filipeflop.com/blog/como-funciona-o-sensor-de-gas-mq-135/>. Acesso em: 17 jan. 2023.
- [6] OLHAR DIGITAL. Raspberry Pi: O que é, para que serve e como comprar. Disponível em: <https://olhardigital.com.br/2019/02/18/noticias/raspberry-pi-o-que-e-para-que-serve-e-como-comprar/>. Acesso em: 16 jan. 2023.
- [7] FILIPEFLOP. Sensor de Umidade e Temperatura DHT11: Disponível em: <https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-dht11/>. Acesso em: 16 jan 2023
- [8] GUIMARÃES, Fábio. Buzzer – Como usar com o Arduino. [S. l.], 11 ago. 2017. Disponível em: <https://mundoprojetado.com.br/buzzer-como-usar-com-o-arduino/>. Acesso em: 18 jan. 2023.

Apêndice

O projeto foi desenvolvido em um único repositório. O código-fonte completo e documentos relacionados podem ser acessados no link a seguir:

- <https://github.com/gabrielaggn/Projeto-SOE>

Sensor de Temperatura e Umidade DHT11 (Python):

```
import Adafruit_DHT
import RPi.GPIO as GPIO
import time

# Define o tipo de sensor
sensor = Adafruit_DHT.DHT11
#sensor = Adafruit_DHT.DHT22

GPIO.setmode(GPIO.BOARD)

# Define a GPIO conectada ao pino de dados do sensor
pino_sensor = 4

# Informacoes iniciais
print ("*** Lendo os valores de temperatura e umidade*");

# Efetua a leitura do sensor
umid, temp = Adafruit_DHT.read_retry(sensor, pino_sensor);
# Caso leitura esteja ok, mostra os valores na tela
if umid is not None and temp is not None:
    with open('dadostempumid.txt', 'w') as arquivo:
        print("Temperatura = {0:0.1f} C Umidade = {1:0.1f} %".format(temp, umid), file=arquivo)
else:
    # Mensagem de erro de comunicacao com o sensor
    print("Falha ao ler dados do DHT11 !!!")
```

Código sensor MQ135 (C):

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <signal.h>
#include <sys/types.h>
#include <unistd.h>

int main(void){
    int alarme = 2; //representa o IO22
    int som = 3; //representa o IO22
    wiringPiSetup();
    pinMode(alarme, INPUT);
    pinMode(som, OUTPUT);

    while(1){
        if(digitalRead(alarme) == LOW){
            //printf("ALERTA!!!");
            system("/home/gabriela/envioAlertaPerigo.sh");
            system("/home/gabriela/Desktop/buzzer");

        }
        else{
            //printf("Tudo okay!");
            /*printf("%d", digitalRead(alarme));*/
            digitalWrite(som, LOW);
            usleep(500);
        }
    }
    return 0;
}
```

Código Buzzer (C):

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <signal.h>
#include <sys/types.h>
#include <unistd.h>

int main(void){

    int som = 3; //representa o IO22
    int alarme = 2; //representa o IO22
    wiringPiSetup();
    pinMode(som, OUTPUT);
    pinMode(alarme, INPUT);

    while(1){

        digitalWrite(som, HIGH);
        usleep(500);
        digitalWrite(som, LOW);
        usleep(500);
        if(digitalRead(alarme) == HIGH){
            digitalWrite(som, LOW);
            break;
        }
    }
}
```

Envio para o telegram

<https://github.com/gabrielaggn/Projeto-SOE/blob/6de25016ae18d465037d1a2553f68327839b55fa/c%C3%B3digo-fonte/envioTelegram.sh>

Envio de Alerta de Perigo

```
#!/usr/bin/env bash

id=$(cat /home/gabriela/Desktop/id.txt)

sudo telegram "$id" "ALERTA, SEU PET CORRE PERIGO"
```