**FUNDAÇÃO ESCOLA TÉCNICA LIBERATO SALZANO VIEIRA DA CUNHA**

**CURSO TÉCNICO DE ELETRÔNICA**

**CAUAN PORTO MACHADO**

**GUILHERME REICHERT DE AGUIAR E LIMA**

**MANUAL DE OPERAÇÃO E MANUTENÇÃO DO DISPOSITIVO DE CAMPAINHA COM CÂMERA WIFI**

**Professor: Leo Weber**

**Novo Hamburgo**

**2024**

# SUMÁRIO

[**SUMÁRIO 1**](#_dw3brznoatxf)

[**1. INTRODUÇÃO 2**](#_rug72fl9xqn0)

[**2. COMPONENTES DO DISPOSITIVO 2**](#_n8bhnk4mcsjw)

[2.1 Raspberry Pi Zero 2 W 2](#_7i3k00yhwz9n)

[2.2 Câmera Raspberry Pi Rev 1.3 2](#_yl3hyb3jwjpd)

[2.3 Botão da campainha 2](#_qlf4wkb3qt38)

[**3. FUNCIONAMENTO 2**](#_sxmu35ab4cdw)

[3.1 Campainha 2](#_lkwywvf4zhnl)

[3.2 Imagem 3](#_a46xku2oeu3g)

[3.3 Notificação 4](#_k9o5powlelyq)

[3.4 Diagrama em blocos 7](#_1rm0yymkq46)

[**4. OPERAÇÃO 8**](#_gxh1jocb9ws4)

[4.1 Requisitos 8](#_c6to2ad5xn2z)

[4.2 Conexão WiFi 8](#_sygzmdcbib0g)

[4.3 Sistema de notificação 8](#_nukkop77fnsr)

[**5. MANUTENÇÃO 9**](#_cutzt9kbphtw)

[5.1 Semanal 9](#_det15cxfdaiu)

[5.2 Mensal 9](#_z8oaaybkb7kl)

[5.3 Anual 9](#_qn427hcpjqoo)

[**6. SOLUÇÃO DE PROBLEMAS 9**](#_edtzbliv64jk)

**SUMÁRIO DE FIGURAS**

[Figura 1 - Página principal 6](#_cj8ms763oxyg)

[Figura 2 - Página secundária 6](#_b1dhjg4jn5f3)

[Figura 3 - Página terciária 7](#_csmhem8h3sah)

[Figura 4 - Diagrama em blocos do dispositivo 8](#_28qjk6h00cgo)

# 1. INTRODUÇÃO

Este manual contém informações detalhadas sobre o dispositivo de campainha com câmera WiFi. O documento disponibiliza especificações técnicas sobre os componentes e o código-fonte do dispositivo, bem como a descrição do seu funcionamento e sua operação.

# 2. COMPONENTES DO DISPOSITIVO

## 2.1 Raspberry Pi Zero 2 W

Um microcomputador desenvolvido pela Raspberry Pi Foundation, com capacidade de processamento de dados, portas GPIO e conexão WiFi. O Raspberry Pi Zero 2 W é o modelo mais compacto da linha Raspberry Pi, possuindo dimensões de 65x30mm.

Para as especificações técnicas, o Raspberry Pi Zero 2 W possui uma memória RAM de 512MB SDRAM, um conector CSI (Camera Serial Interface), um slot para cartão micro SD de até 64GB e um processador ARM Cortex-A53 Quad-Core 64 bits de 1GHz.

## 2.2 Câmera Raspberry Pi Rev 1.3

Uma câmera com conector CSI (Camera Serial Interface) compatível com Raspberry Pi. Possui resolução Full HD, ângulo de visão de 54º, e dimensões de 25x24mm. Seu controle é feito em código Python através da biblioteca “Picamera2”.

## 2.3 Botão da campainha

Um botão utilizado como campainha no dispositivo.

# 3. FUNCIONAMENTO

A campainha com câmera WiFi tem como objetivo facilitar a visualização e verificação via celular da entrada de um local. O processo de funcionamento do equipamento pode ser dividido em 4 etapas:

* Etapa 1: Pressionamento da campainha
* Etapa 2: Captura de imagem
* Etapa 3: Envio da notificação
* Etapa 4: Recebimento da notificação

No momento em que o visitante pressiona a campainha, um sinal elétrico é enviado para o microcomputador, que interpreta o sinal recebido e inicia o protocolo de captura de imagem. A imagem capturada é enviada para o servidor via WiFi, com destino ao celular do dono do local onde a campainha foi pressionada. O celular do dono recebe uma notificação em seu celular da chegada do visitante juntamente com sua imagem, permitindo que o dono seja capaz de ver previamente quem pressionou a campainha do seu estabelecimento ou moradia.

## 

## 3.1 Campainha

O sinal elétrico emitido pelo pressionamento do botão da campainha é capturado pelo microcomputador através de suas portas GPIO. A configuração dessas portas é feita utilizando a biblioteca “RPi.GPIO”, apelidada no programa de “GPIO”. A função “GPIO.setmode” configura a referência utilizada para configurar o número dos pinos GPIO, enquanto a função “GPIO.setup” configura o pino determinado como saída através do parâmetro “GPIO.IN”. O botão da campainha é dimensionado com um resistor de Pull-Up, de modo que seu acionamento se dá em nível baixo (Low - 0V).

# Configurando GPIO Campainha

# Pull-up físico / Aciona em Low

GPIO.setmode(GPIO.BCM)

PinoCampainha = 23 # Pino GPIO 23

GPIO.setup(PinoCampainha, GPIO.IN)

O estado do botão é verificado constantemente através de um laço de repetição infinito no programa. O estado do botão é capturado e informado pela função “GPIO.input”, tendo como referência o estado Low (0V) para a indicação de pressionamento da campainha. No momento em que o código detecta o pressionamento do botão, as funções de capturar a imagem e enviar a notificação são chamadas, realizando os procedimentos de funcionamento do dispositivo. Após a conclusão dos procedimentos, uma contagem de 10 segundos é iniciada, de modo que nenhuma detecção de pressionamento do botão é realizada neste período, visando evitar problemas com spam.

# Loop principal

try:

while True:

if not GPIO.input(PinoCampainha):

print("Campainha Pressionada!")

FotoCampainha()

EnviaNotificacao()

# Impede spam

time.sleep(10)

finally:

if picam2:

picam2.stop()

print("Câmera desligada.")

GPIO.cleanup()

## 3.2 Imagem

A captura de imagem é feita através de uma rotina de código em Python, utilizando a biblioteca “Picamera2” apelidada de “picam2” no programa, sendo uma ferramenta que permite o controle e configuração da câmera Raspberry. A câmera é inicializada através da função “picam2.start()” da biblioteca, tendo seus possíveis erros tratados através do método “try/except” e comunicados para o usuário.

picam2 = Picamera2() # Facilita configuração da câmera

# Inicializando a câmera

try:

picam2.start()

print("\n\n\n\n\n\nCâmera inicializada com sucesso!")

except Exception as erro:

print(f"Erro ao inicializar a câmera: {erro}")

picam2 = None

A captura da imagem é feita através da função “Foto Campainha()” declarada no programa. A função “picam2.capture\_file” captura a imagem da câmera e armazena o arquivo. O local de armazenamento é passado como parâmetro para a função através de uma string que contém todo o caminho percorrido entre os arquivos até a imagem capturada, em formato “.jpg”. Novamente, possíveis erros são tratados através do método “try/except” e comunicados ao usuário.

# Função para capturar foto

def FotoCampainha():

if picam2:

try:

picam2.capture\_file

('/home/gaiola/Desktop/ImagemCampainha/FotoCampainha.jpg')

print('Foto capturada!')

except Exception as e:

print(f"Erro ao capturar a foto: {e}")

## 3.3 Notificação

O sistema de notificação é feito através do serviço Pushover, uma ferramenta de envio e recebimento de notificações para Android e iOS. O aplicativo conta com uma página principal, onde é possível visualizar todo o histórico de mensagens (notificações) recebidas. Ao tocar em uma mensagem, uma nova página é apresentada, mostrando a imagem anexada à mensagem. Tocando na imagem, mais uma página é disponibilizada, contendo dados precisos de data e hora em que a notificação foi recebida, bem como opções de compartilhamento, salvamento e remoção da notificação.

### Figura 1 - Página principal

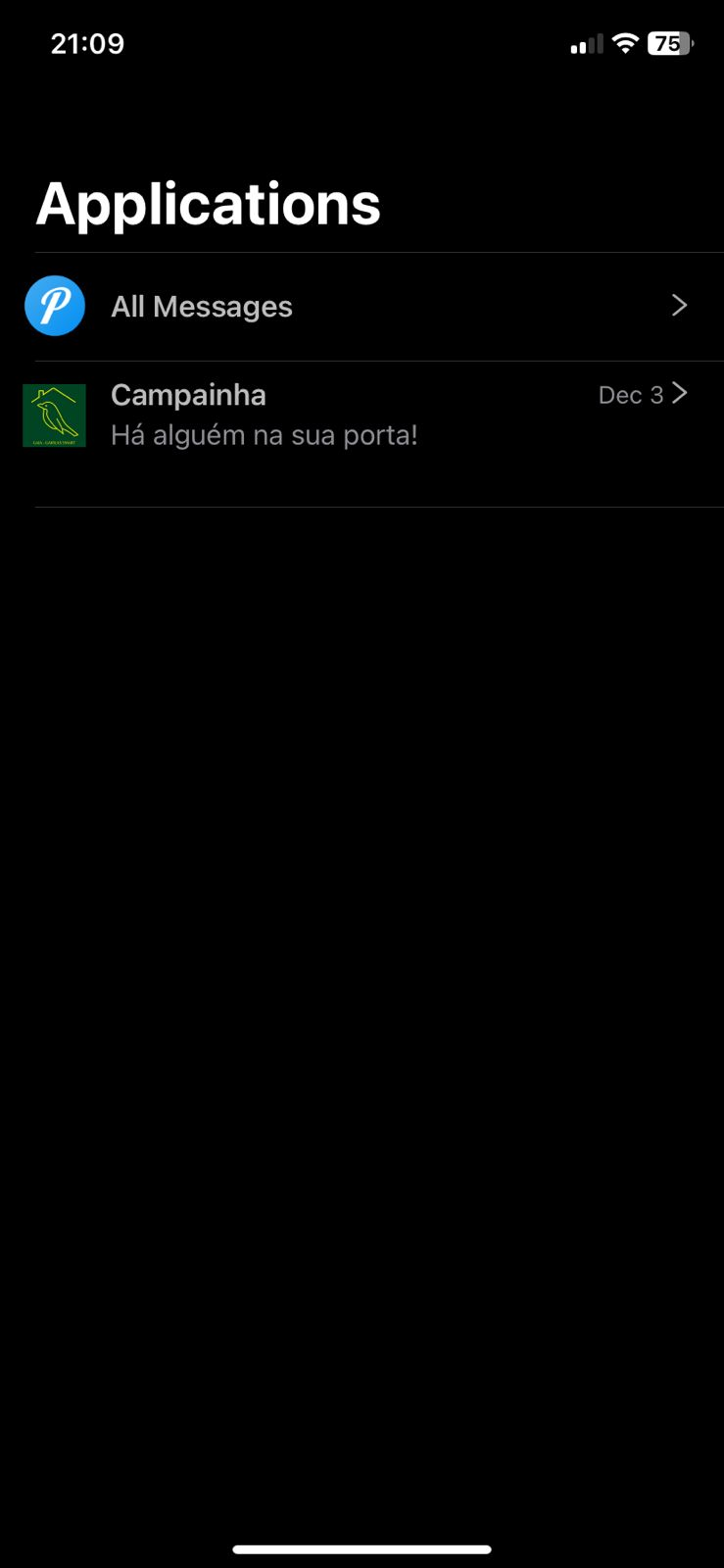
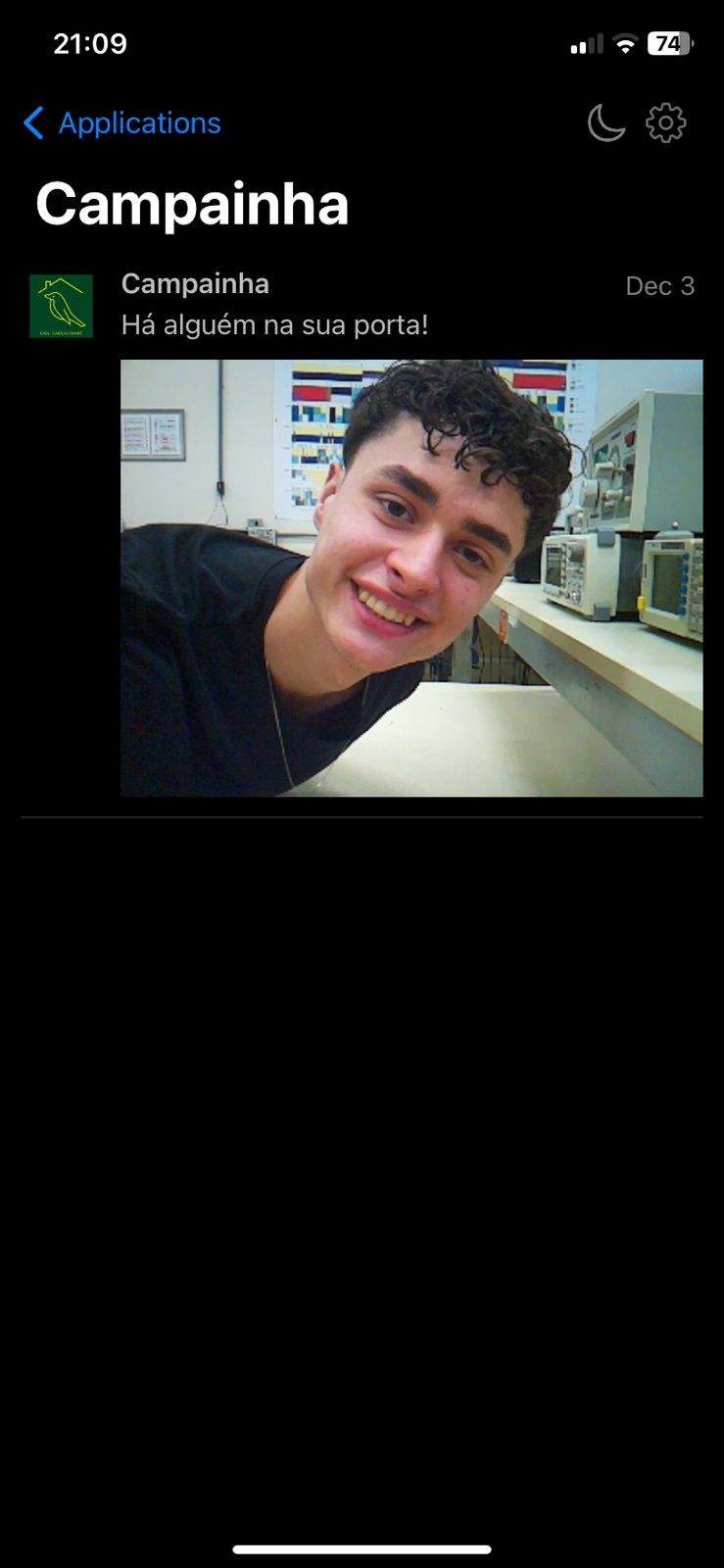
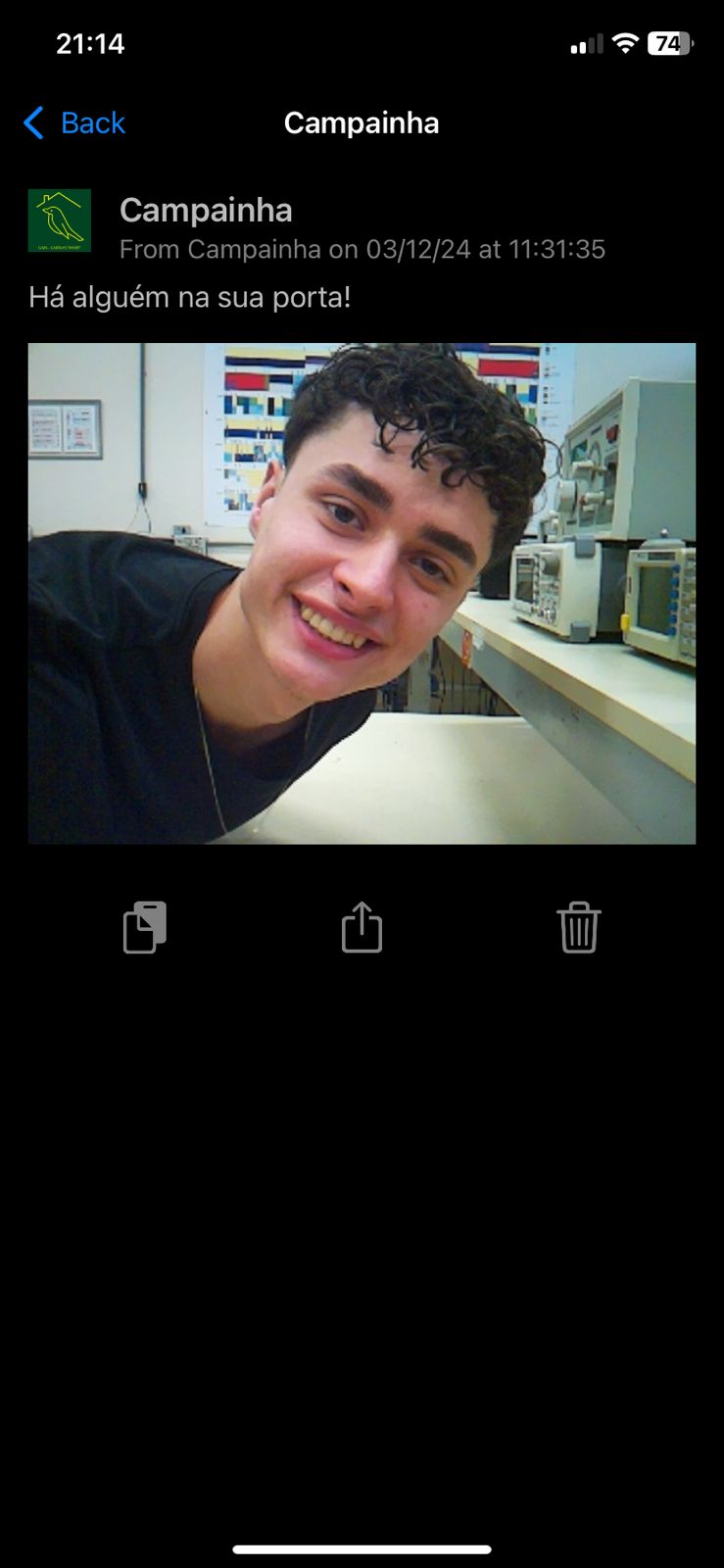


Figura 2 - Página secundária



### Figura 3 - Página terciária



As notificações são criadas e enviadas através de uma rotina de código em Python compilada dentro do Raspberry Pi Zero 2 W. A biblioteca “requests” foi utilizada para realizar o envio das notificações, e a biblioteca “random” para sortear a mensagem enviada para o usuário. As duas primeiras variáveis da função “EnviaNotificacao()” são as chaves de acesso, que garantem o endereçamento correto da notificação. A variável “imagem\_path” contém o caminho percorrido entre os arquivos para acessar a foto capturada. A variável “url” possui o endereço do link em que o arquivo contendo a mensagem, a imagem e o destinatário é enviado através do método “post”, onde é convertido e formatado pelo serviço Pushover e enviado para o celular do usuário.

# Função para enviar notificação

def EnviaNotificacao():

# Configurações de Acesso

api\_token = "API\_TOKEN" # Chave de API

user\_key = "USER\_KEY" # Chave de Usuário

# Configurações de Mensagem de Texto

mensagens = ["Há alguém na sua porta!",

"A campainha tocou!",

"Sua visita chegou!"]

mensagem = mensagens[random.randint(0, 2)]

# Caminho da imagem capturada

imagem\_path =

"/home/gaiola/Desktop/ImagemCampainha/FotoCampainha.jpg"

# Envio da Notificação

url = "https://api.pushover.net/1/messages.json"

data = {

"token": api\_token,

"user": user\_key,

"message": mensagem,

}

try:

with open(imagem\_path, "rb") as image\_file:

files = {"attachment": image\_file}

response = requests.post(url, data=data, files=files)

if response.status\_code == 200:

print("Notificação enviada com sucesso!")

else:

print(f"Erro ao enviar notificação: {response.status\_code}, {response.text}")

except FileNotFoundError:

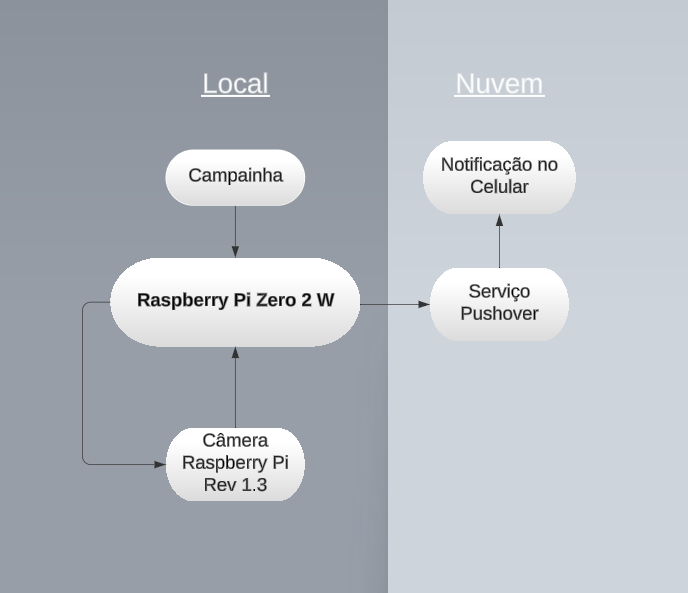
print("Erro: Imagem não encontrada para envio.")

## 

## 3.4 Diagrama em blocos

O diagrama em blocos oferece um panorama geral do funcionamento do dispositivo de forma resumida.

### Figura 4 - Diagrama em blocos do dispositivo



# 4. OPERAÇÃO

Neste capítulo serão abordados os procedimentos necessários para a operação adequada do dispositivo de campainha com câmera WiFi.

## 4.1 Requisitos

Alguns materiais específicos são necessários para a instalação do dispositivo, bem como seu manuseio e operação.

Periféricos básicos de computador: Monitor, mouse e teclado são utilizados para configuração do microcomputador Raspberry Pi Zero 2 W.

* Acesso à Internet: Uma conexão WiFi estável para realizar um cadastro online.
* Celular Android ou iOS: Instrumento que receberá as notificações de visitas enviadas pelo sistema de campainha.
* Fonte 5V/3A: Alimentação do microcomputador que realiza o controle do dispositivo.

## 4.2 Conexão WiFi

Para realizar a configuração do sistema de campainha é necessário acessar o Raspberry Pi Zero 2 W utilizando os periféricos, partindo do pressuposto de que o sistema operacional Raspibian já está instalado no microcomputador, bem como as bibliotecas utilizadas no código-fonte da campainha.

Com a interface gráfica do microcomputador aberta para a visualização no monitor, estabelecer uma conexão WiFi inserindo a senha da rede utilizando mouse e teclado. Essa rede WiFi deve ser estável e próxima ao dispositivo de campainha para garantir o funcionamento adequado.

## 4.3 Sistema de notificação

Após criar uma conta gratuita no site “pushover.net” e realizar o login, é possível obter, na aba “Dashboard” do site, a “User Key”, uma chave de acesso vinculada ao seu usuário utilizada para validação da notificação enviada.

O celular que receberá as notificações deve ter o aplicativo Pushover instalado, onde deve ser realizado o cadastro da mesma conta utilizada previamente no site do serviço. Na aba “Create an Application/API Token” do site, é possível criar a “API Token”, uma chave de acesso utilizada para direcionar corretamente a notificação para determinado aparelho.

Com a “User Key” e a “API Token” em mãos, basta acessar o código-fonte da campainha no Raspberry Pi por meio de periféricos e substituir as lacunas “API\_TOKEN” e “USER\_KEY”, mantendo as aspas, pelas chaves de acesso adquiridas anteriormente.

# Configurações de Acesso

api\_token = "API\_TOKEN" # Chave de API

user\_key = "USER\_KEY" # Chave de Usuário

Deste modo, a comunicação entre o sistema de campainha e o celular do usuário está estabelecida e as notificações podem começar a ser enviadas.

# 5. MANUTENÇÃO

## 5.1 Semanal

Verificar e inspecionar câmera e demais componentes, buscando possíveis danos ou sujeiras.

## 5.2 Mensal

Limpar a lente da câmera com papel higiênico ou papel toalha, juntamente com álcool de concentração igual ou superior a 70%.

## 5.3 Anual

Encaminhar o dispositivo para um técnico especializado inspecionar as conexões e realizar as manutenções e limpezas necessárias.

# 6. SOLUÇÃO DE PROBLEMAS

| **Problema** | **Causa Possível** | **Solução** |
| --- | --- | --- |
| Falha ao inicializar a câmera: “Failed to acquire camera: Device or resource busy”. | Câmera está sendo utilizada por outro programa. Câmera está ocupada. | Certificar-se que nenhum outro programa está mantendo a câmera ocupada. Reinicar o sistema. |
| Mensagem “Campainha pressionada!” não é exibida no Shell. | Conexão da campainha não está correta. Pino GPIO da campainha é diferente do pino configurado no código. | Verificar conexão elétrica. Utilizar o mesmo pino configurado no código para conexão da campainha. |
| Erro ao enviar notificação | Endereçamento incorreto. User Key ou API Token do código-fonte diferente da User Key e da API Token esperada pelo Pushover. | Verificar se a User Key e a API Token inseridas no código-fonte são idênticas às fornecidas pelo serviço Pushover. |
| Erro ao receber notificação no celular | Falha na conexão com a Internet no celular | Verificar conexão com a Internet no celular, inspecionando estabilidade. |