



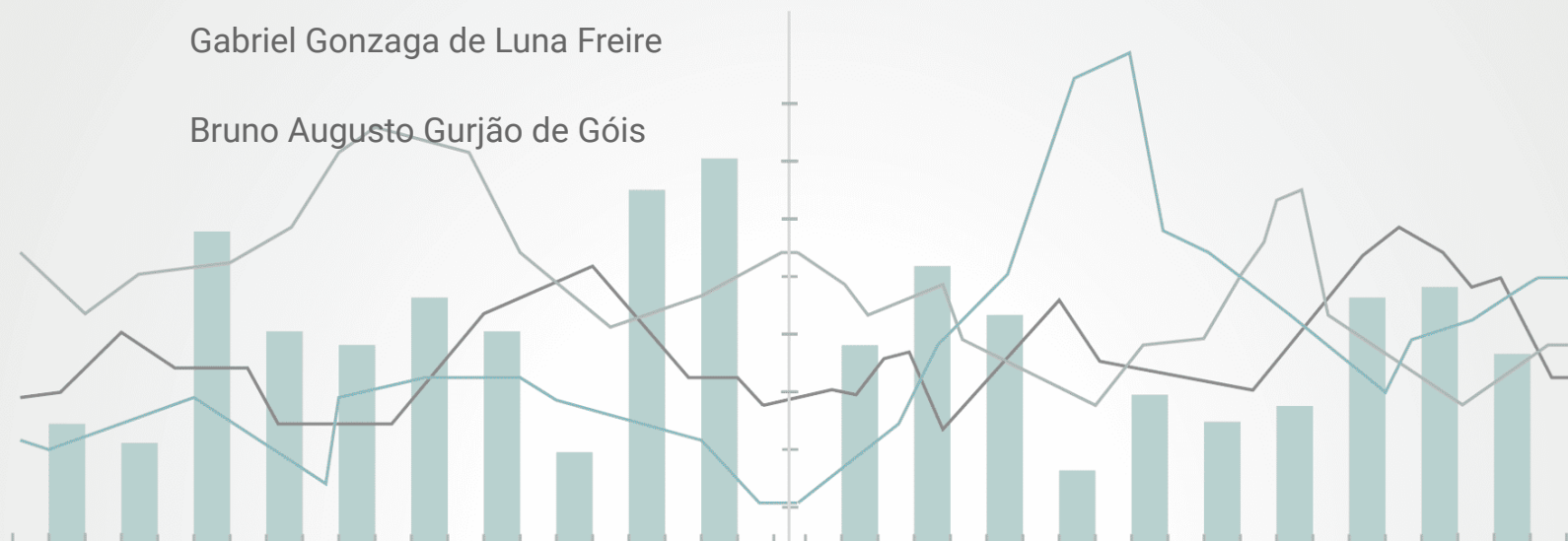
## Relatório de Projeto de Internet das Coisas (IoT)

### Identificação de presença de pessoas usando visão computacional

Magdiel Gomes Ferreira

Gabriel Gonzaga de Luna Freire

Bruno Augusto Gurjão de Góis



# Controle de temperatura de acordo com a quantidade de pessoas

Disciplina: IMD0902 - Introdução a Internet das Coisas - 2023.2

Orientador: Prof. Heitor Medeiros Florencio

Prof. Leonardo Augusto

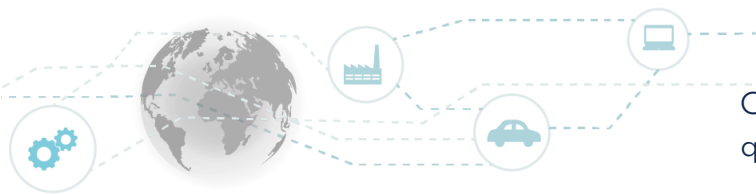
Universidade Federal do Rio Grande do Norte

## Descrição do projeto

O projeto trabalhado na terceira unidade de Introdução a Internet das Coisas foi criado com objetivo de identificar pessoas através de visão computacional. Para isso foram usados equipamentos como Raspberry Pi 4 e ESP32-CAM. Esses dois dispositivos trabalharão juntos para identificar rosto e enviar mensagens de avisos de acordo com as condições que serão mencionadas neste relatório.

## Resumo

Ao usar o ESP32-CAM, imagens serão enviadas através do ip para o Raspberry Pi 4 e neste as imagens serão processadas, trabalhadas usando biblioteca cv2 (Opencv) e a biblioteca de identificação de rosto (mediapipe), e, assim, ao detectar um rosto ou deixar de reconhecê-lo, mensagens serão enviadas tanto para o Telegram usando a biblioteca telebot, quanto para o Adafruit IO usando a biblioteca Adafruit\_IO.



Controle de temperatura de acordo com a quantidade de pessoas

## 1. Introdução

Este projeto foi pensado para ser implementado na recepção do prédio no nPITI (UFRN). A problemática identificada é não saber se há alguém na recepção para poder pegar a chave de uma sala do próprio nPITI. Portanto, este trabalho procura facilitar esse processo usando equipamentos (Raspberry e ESP32-CAM) e uma lógica programada na linguagem python.

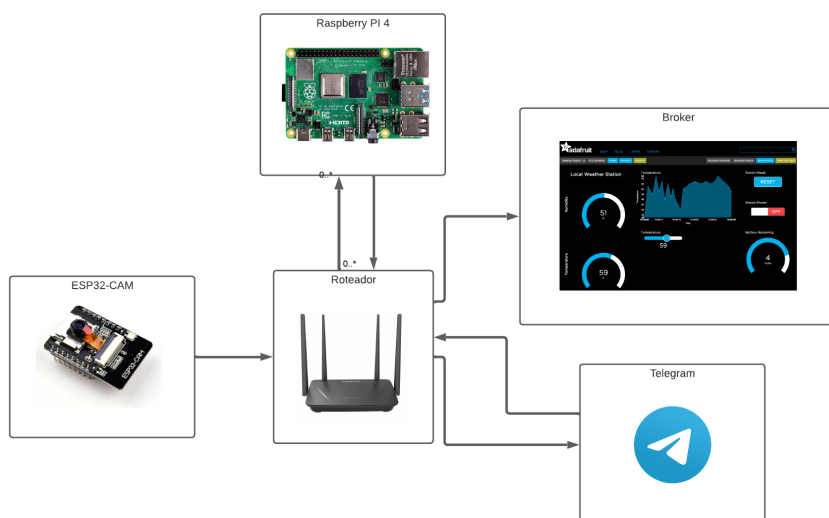
Através do ESP32-CAM enviaremos imagens, usando o seu IP, para o Raspberry Pi 4 e nele, usando o código programado em python identificaremos se há alguém ou não na recepção. Ao não detectar a presença de alguém por um determinado tempo, uma mensagem será enviada para o Telegram e para o Adafruit IO.

Pelo broker Adafruit IO podemos ver quanto tempo a pessoa não foi identificada, o tempo de saída, o tempo da volta e a data em que isso ocorreu. Caso queiramos ver esses valores no próprio dispositivo, os dados enviados para o broker também são armazenados em arquivo no próprio Raspberry.

Outro detalhe acrescentado neste projeto é saber se realmente a pessoa está ou não na recepção fazendo a requisição de uma foto através do telegram, por isso, para receber mensagens ou fazer a requisição é necessário estar no grupo criado.

## 2. Arquitetura IoT

Logo abaixo representamos a arquitetura do projeto.



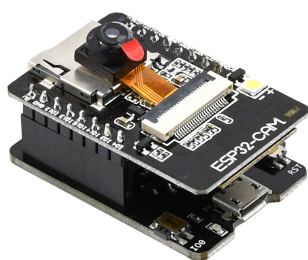


**Figura 01** - Exemplo de arquitetura de projeto IoT. Fonte: Criado na plataforma Lucidchart.

ESP32-CAM enviará imagens para o Raspberry, que por sua vez enviará informações para o Telegram e para o Adafruit IO. Do Telegram pode-se enviar requisição de foto para o Raspberry.

Logo a seguir serão apresentados os equipamentos.

### **ESP32-CAM**



**Figura 02:** ESP32-CAM. Fonte: Amazon. Disponível em:

<<https://www.amazon.com.br/Aideepen-Bluetooth-ESP32-CAM-ESP32-CAM-MB-Arduino/dp/B08P2578LV?th=1>>. Acesso em: 11 de dezembro de 2023.

Esta placa integra o chip ESP32 e uma câmera. Além disso, há entrada para cartão SD e um LED funcionando como flash. Este módulo é capaz de fazer stream de vídeo, tirar fotos e processar reconhecimento facial.

### **Raspberry PI 4 Model B**





Controle de temperatura de acordo com a quantidade de pessoas

**Figura 03:** Raspberry Pi 4 Model B. Fonte: Robocore. Disponível em:

<<https://www.robocore.net/placa-raspberry-pi/raspberry-pi-4-8gb>>. Acesso em: 11 de dezembro de 2023.

É um microcomputador completo, com seus componentes em um única placa lógica. Há o processador, a memória RAM e a placa de vídeo impressos, e entradas USB, HDMI, áudio e vídeo composto, para câmera e telas LCD e uma GPIO. A alimentação é feita através de uma porta microUSB.

### 3. Metodologia

Para cumprir os requisitos propostos, realizamos o processamento dos dados do ESP32-CAM no dispositivo IoT (Raspberry Pi 4) e para gerar dashboards e outras funcionalidades usamos o Adafruit IO e o Telegram.

#### 3.1. Dispositivos IoT

Os dispositivos IoT já mencionados anteriormente foram o ESP32-CAM e o Raspberry Pi 4.

Para programar a ESP32-CAM usamos a IDE Arduino e nessa IDE é necessário adicionar o tipo da placa, que no nosso caso é a AI Thinker ESP32-CAM. Nesse IDE já existe o modelo para ESP32-CAM, porém precisamos fazer alguns ajustes como, por exemplo, colocar o SSID e a senha da rede.



Controle de temperatura de acordo com a quantidade de pessoas

```
1  #include "esp_camera.h"
2  #include <WiFi.h>
3
4  //
5  // WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
6  //           Ensure ESP32 Wrover Module or other board with PSRAM is selected
7  //           Partial images will be transmitted if image exceeds buffer size
8  //
9
10 // Select camera model
11 //define CAMERA_MODEL_WROVER_KIT // Has PSRAM
12 //define CAMERA_MODEL_ESP_EYE // Has PSRAM
13 //define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
14 //define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
15 //define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
16 //define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
17 #define CAMERA_MODEL_AI_THINKER // Has PSRAM
18 //define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
19
20 #include "camera_pins.h"
21
22 const char* ssid = "*****";
23 const char* password = "*****";
24
25 void startCameraServer();
26
```

**Figura 04** - Modelo do ESP32-CAM. Fonte: Arduino. .

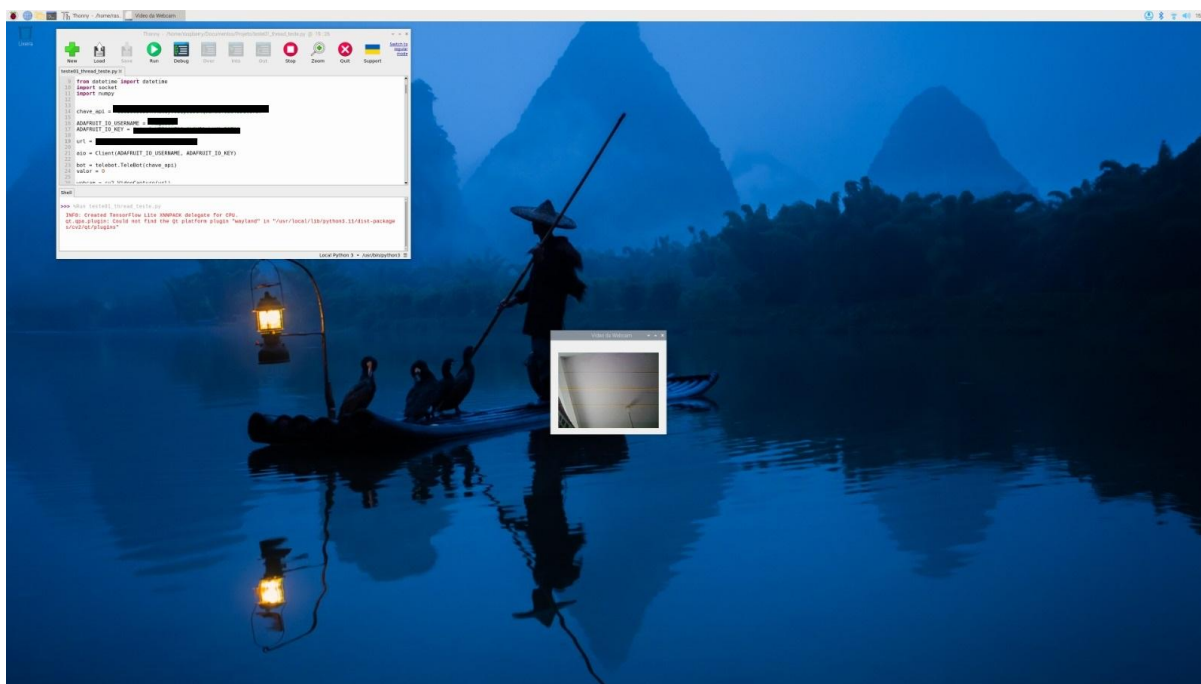
Para fazer a gravação do código no módulo ESP32-CAM, precisamos usar um cabo micro usb. Logo depois de fazer a gravação observamos no monitor serial da IDE do arduino qual o ip e acessamos para capturar a imagem.

Colocamos o ip para o opencv fazer o processamento da imagem capturada pelo IP.

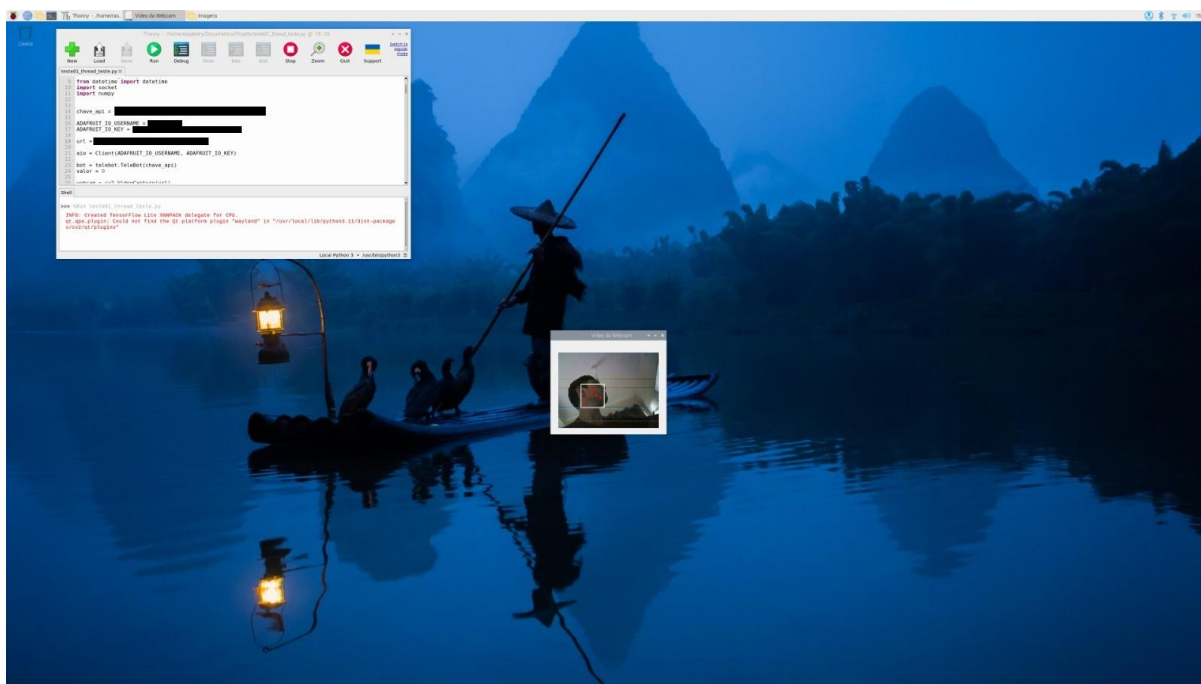




Controle de temperatura de acordo com a quantidade de pessoas



**Figura 05** - Usando o Raspberry Pi 4. Fonte: Própria.



**Figura 06** - Usando o Raspberry Pi 4. Fonte: Própria.

Se por acaso o Raspberry Pi 4 não conseguir capturar o rosto em um intervalo de 10 segundos, é enviada a mensagem de que não há ninguém na recepção.



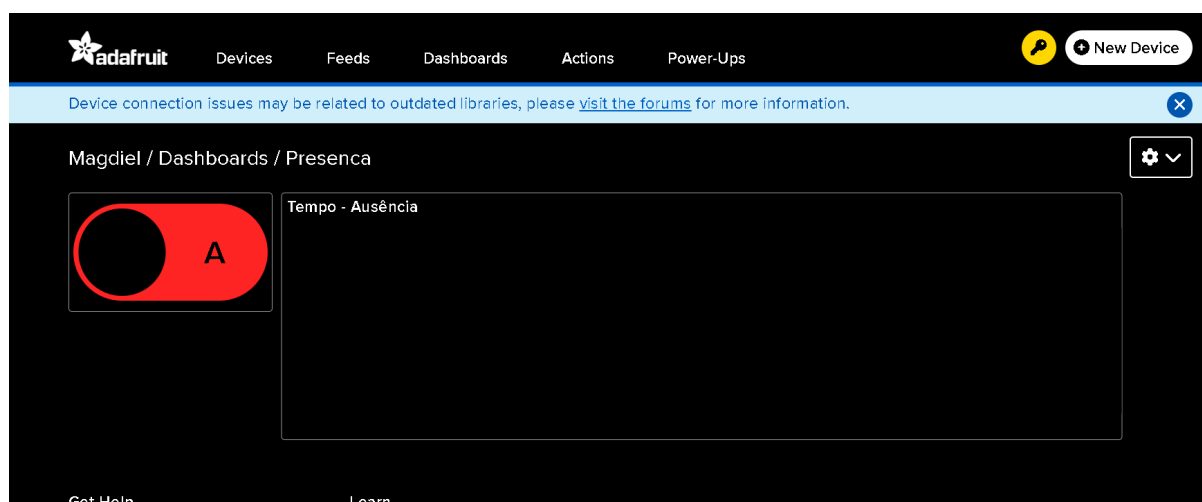
Controle de temperatura de acordo com a quantidade de pessoas

### 3.2. Conectividade

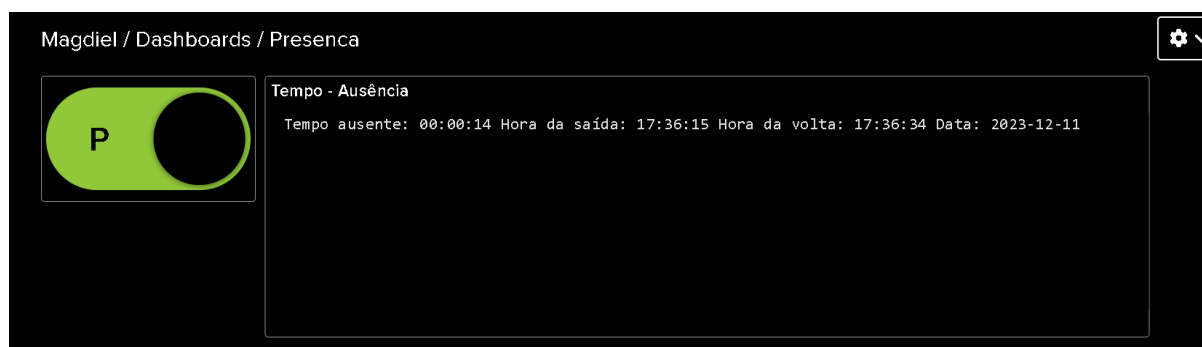
Para fazer a conexão entre os dispositivos e para estarem em uma mesma rede, optamos por colocar na rede 4G do celular porque foi vista uma maior rapidez na captura das imagens e um menor delay. Para o ESP32-CAM usamos a biblioteca WiFi.h para fazer a conexão com a internet.

### 3.3. Plataformas, Aplicações e Serviços

A plataforma usada no projeto foi o Adafruit IO. Essa é uma plataforma de IoT que recebe dados através dos feeds e por meio destes podemos tornar a visualização das informações mais fácil usando dashboards. O Adafruit IO recebe informações de dispositivos e envia para estes através do protocolo MQTT. Cada informação terá o seu feed, porém podemos unir os dados em uma única dashboard. A seguir, são apresentados os exemplos.



**Figura 7** - Exemplo de dashboard. **Fonte:** Adafruit IO.



**Figura 8** - Exemplo de dashboard. **Fonte:** Adafruit IO.





Controle de temperatura de acordo com a quantidade de pessoas

Para fazer a publicação no feed, precisa-se definir o nome do usuário, a chave. Isso é feito no código do Raspberry Pi 4.

```
from Adafruit_IO import RequestError, Client, Feed
from datetime import datetime
import socket
import numpy

chave_api = [REDACTED]

ADAFRUIT_IO_USERNAME = [REDACTED]
ADAFRUIT_IO_KEY = [REDACTED]

url = [REDACTED]

aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)
```

**Figura 9** - Publicação de dados em um feed.

Para fazer a publicação pelo python é necessário importar a biblioteca Adafruit\_IO.

## 4. Resultados e discussões

Como queríamos algum projeto que envolvesse visão computacional inicialmente existiram dúvidas a respeito, mas ao conversar com o professor e orientador Leonardo Augusto tomamos a decisão de identificar a presença ou não de uma pessoa e enviar mensagem para Adafruit IO e para o telegram.

Testamos o projeto em Raspberry Pi 2, porém, devido a não conseguirmos instalar a biblioteca opencv nesse dispositivo, optamos por usar um dispositivo um pouco mais atual, ou seja, o Raspberry Pi 4 Model B.

Antes de possuímos o dispositivo, manipulamos no computador usando o pycharm. No computador a requisição de tirar foto funcionou, porém ao adaptarmos



Controle de temperatura de acordo com a quantidade de pessoas

para o Raspberry Pi 4, em um momento funcionou, mas nos outros casos, não, o que pretendemos futuramente consertar este erro.

Um outro problema que precisamos contornar foi usar o opencv e o bot do telegram ao mesmo tempo. Isso contornamos usando thread com a biblioteca threading, para que as funções ocorram paralelamente.

Logo abaixo há um link para ver o funcionamento do projeto:

<https://youtu.be/HHclAGAy2qA>



Controle de temperatura de acordo com a quantidade de pessoas

## 5. Referências

[1] HENRIQUE, Eduardo. Título: Introdução ao ESP32-CAM. Blog Eletrogate. Disponível em: <<https://blog.eletrogate.com/introducao-ao-esp32-cam/>> . Acesso em: 11/12/2023.

[2] GOGONI, Ronaldo. Título: O que é Raspberry Pi?. Tecnoblog. Disponível em: <<https://tecnoblog.net/responde/o-que-e-o-raspberry-pi/>> . Acesso em: 11/12/2023.

[4] ZUQUETO, Douglas. Título: Adafruit IO - Uma nova plataforma de IoT. Maker Hero. Disponível em: <<https://www.makerhero.com/blog/adafruit-io-plataforma-iot/>>. Acesso em: 30/10/2023.