

Dispositivo para controle e segurança de um profissional via Arduino, aplicativo Android e Google Firebase

Guilherme Schurgelies Kishi

**Dispositivo para controle e segurança de um
profissional via Arduino, aplicativo Android e Google
Firebase**

Aluno(s): Guilherme Schurgelies Kishi

Orientador(es): Jan Krueger Siqueira

Trabalho apresentado com requisito parcial à conclusão do curso de Engenharia Elétrica na Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

Agradecimentos

Agradeço a minha falecida mãe pelo apoio e carinho,
aos meus amigos que são muitos
e ao meu querido pai pela graduação.

Resumo

Neste presente projeto, foi feito um monitoramento de um indivíduo com o uso de câmera, sensor cardíaco, sensor de som e GPS. Os três primeiros dados são coletados por um microcontrolador chamado ESP32-CAM, enviados de forma sequencial por Bluetooth e coletados por um aplicativo Android. Os dados são processados pelo smartphone, exibidos na tela e enviados para o Google Firebase. Por fim, uma página web carrega e exibe o histórico dessas informações.

Palavras-chave: sensor; Arduino; aplicativo Android; Bluetooth; ESP32; Firebase; GPS.

Device for professional control and security via Arduino, Android app and Google Firebase

Abstract

In this present project, an individual was monitored using a camera, cardiac sensor, sound sensor and GPS. The first three data are collected by a microcontroller called ESP32-CAM, sent sequentially over Bluetooth and collected by an Android application. Data is processed by the smartphone, displayed on the screen and sent to Google Firebase. Finally, a web page loads and displays the history of this information.

Keywords: sensor; Arduino; app Android; Bluetooth; ESP32; Firebase; GPS.

Lista de ilustrações

Figura 1 – Desenho do sistema do processador	2
Figura 2 – Desenho do chip	2
Figura 3 – Desenho esquemático do ESP32	4
Figura 4 – Placa FTDI usada	5
Figura 5 – Sensor de batimento cardíaco	5
Figura 6 – Fronte do sensor	6
Figura 7 – Reflexão da luz verde no sangue	6
Figura 8 – Sensor de som	7
Figura 9 – Gráfico do batimento cardíaco + ruído do ambiente.....	8
Figura 10 – Captura de imagem e processamento	10
Figura 11 – Página inicial e secundária do aplicativo	11
Figura 12 – Storage do Firebase.....	12
Figura 13 – Firestore do Firebase	12
Figura 14 – Web page com os dados do Firebase	13
Figura 15 – Fonte para protoboard	13

Sumário

1. Introdução	
a. Internet das coisas	1
b. Uso de câmeras e monitoramento cardíaco.....	1
c. ESP32.....	1
2. Implementação do projeto	
a. Componentes	4
I. ESP32-CAM	4
II. Módulo FTDI	5
III. Sensor cardíaco	5
IV. Sensor de som	6
b. Ciclo de dados do sistema	7
c. Gráfico do sensor de batimento cardíaco e sensor de som.....	8
d. Cálculo do BPM	8
e. Cálculo da qualidade do som	9
f. Captura e envio da foto	9
g. Envio de foto por WIFI	9
h. Envio de foto por Bluetooth	9
i. OpenCV	10
j. Aplicativo Android	10
k. Firebase e página web	11
l. Alimentação	13
m. Ambientes de desenvolvimentos	14
3. Conclusão	15
4. Bibliografia	16

1. Introdução

a. Internet das coisas

A “Internet das Coisas” se refere a uma revolução tecnológica que tem como objetivo conectar os itens usados do dia a dia à rede mundial de computadores. Cada vez mais surgem eletrodomésticos, meios de transporte e até mesmo tênis, roupas e maçanetas conectadas à Internet e a outros dispositivos, como computadores e smartphones [9].

a. 1) Como surgiu

A ideia de conectar objetos é discutida desde 1991, quando a conexão TCP/IP e a Internet que conhecemos hoje começou a se popularizar. Bill Joy, cofundador da Sun Microsystems, pensou sobre a conexão de Device para Device (D2D), tipo de ligação que faz parte de um conceito maior, o de “várias webs”[9]. Em 1999, Kevin Ashton do MIT propôs o termo “Internet das Coisas” e dez anos depois escreveu o artigo “A Coisa da Internet das Coisas” para o “RFID Journal”. De acordo com o especialista, a rede oferecia, na época, 50 Pentabytes de dados acumulados em gravações, registros e reprodução de imagens [9]. Segundo Ashton, com as limitações de tempo que se tem hoje, as pessoas procurarão se conectar com a internet de outras maneiras, sendo possível acumular dados do movimento do corpo com precisão maior. Sendo possível, portanto, otimizar e economizar recursos naturais e energéticos, proporcionando grandes revoluções.

b. Profissões e situações que tem se beneficiado do uso de câmeras e monitoramento cardíaco

b. 1) Policiais com câmera

De acordo com o site de notícias UOL de 13/07/2021: Não houve registro de morte nas operações dos 18 batalhões da Polícia Militar de São Paulo que fazem parte do programa de uso de câmeras corporais. Os equipamentos são fixados na farda dos policiais e fazem registro contínuo de áudio e vídeo. Ao todo, são mais de 3 mil equipamentos em operação, sendo 585 adquiridos no ano passado e 2,5 mil comprados e distribuídos neste ano. O contrato, que prevê ainda o monitoramento da localização dos policiais e a disponibilização das informações em nuvem, tem um investimento de cerca de R\$ 1,2 milhão por mês [10]. De acordo com o governo de São Paulo na época, as câmeras ajudam a garantir os direitos individuais dos cidadãos durante as abordagens e ações policiais, assim como preservam a transparência das operações e contribuem para produção de provas judiciais [10].

b. 2) Monitoramento de crianças

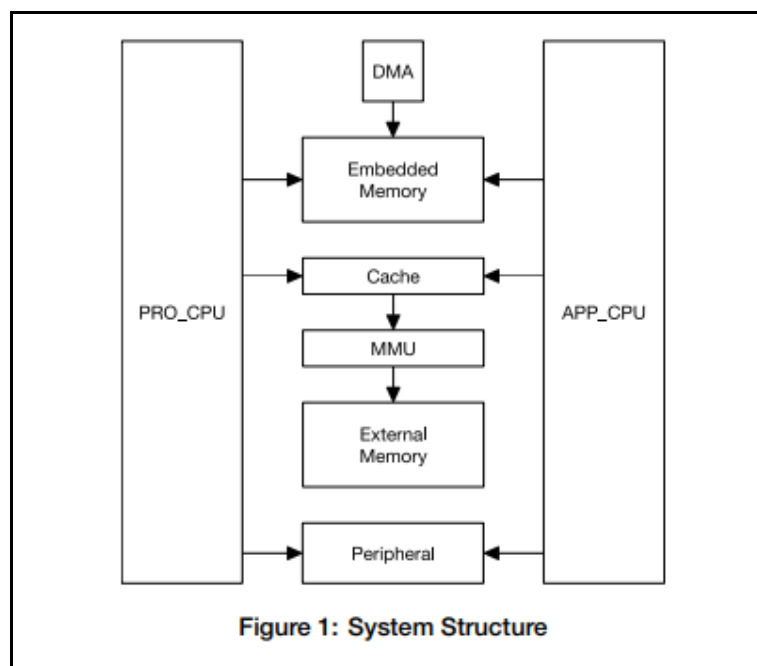
É possível imaginar que o uso de câmeras e monitor cardíaco em crianças, possa de fato trazer grandes resultados para a prevenção de acidentes principalmente, reduzindo o estresse dos pais na sua vigília diária e aumentando a confiança das crianças sabendo que estão sendo monitoradas. Tal como podemos observar no uso de “babas eletrônicas”, muito comum neste ano de 2021 devido a sua ampla variedade de funções que vão além das câmeras, tais como canção de ninar, audiobooks, temperatura ambiente, tracker e microfone, um bom exemplo é o modelo da Motorola Confort85Connect [11].

c. ESP32

Microcontrolador (SoC: system on chip): são chips com baixo poder de processamento em comparação com os microprocessadores. Com baixo consumo de energia, entradas e saídas embutidas no próprio chip, além de outros periféricos tais como Bluetooth, wifi, interfaces de comunicação (serial, USB), uma pequena memória RAM e EPROM (para programas e dados e circuitos de oscilador(clock)) tudo no mesmo chip.[15]

No caso do ESP32 a estrutura de seu sistema é mostrada abaixo:

Figura 1: Desenho do sistema do processador

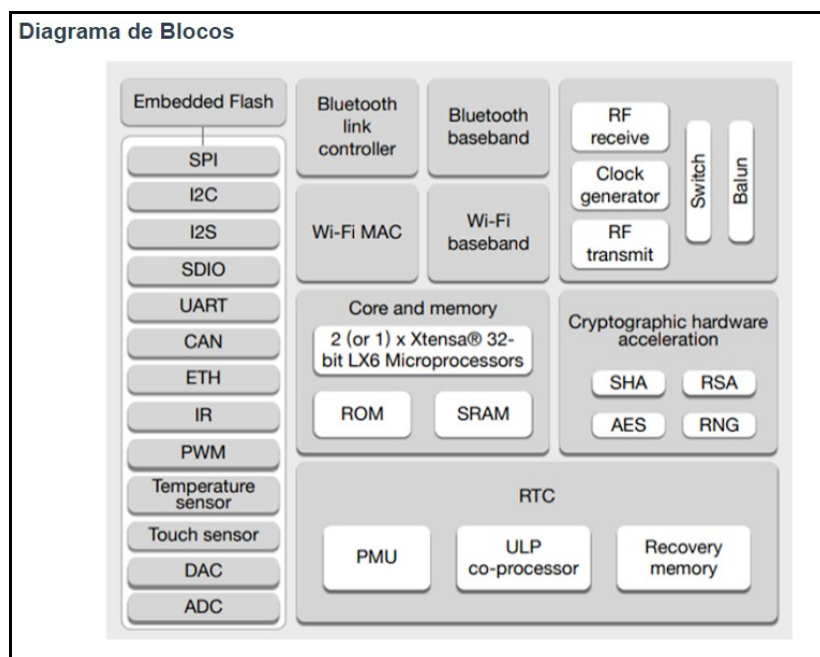


Fonte: Expressif.com

Como mostrado acima, possui duas CPU's, memória externa, periféricos (41 entradas e saídas que são específicas tais como para GPIOs, controlador PID, Secure Boot e etc) e controladores DMA. [16]

Abaixo segue o desenho do chip com seus diversos componentes:

Figura 2: Desenho do chip



Fonte: fernandok.com



O ESP32 também possui GPIO's que são um tipo de pino encontrado em circuito integrado que não tem uma função específica. Enquanto muitos pinos têm uma função específica, como enviar um sinal para um componente, a função do GPIO é customizável e pode ser controlada por software.[12]

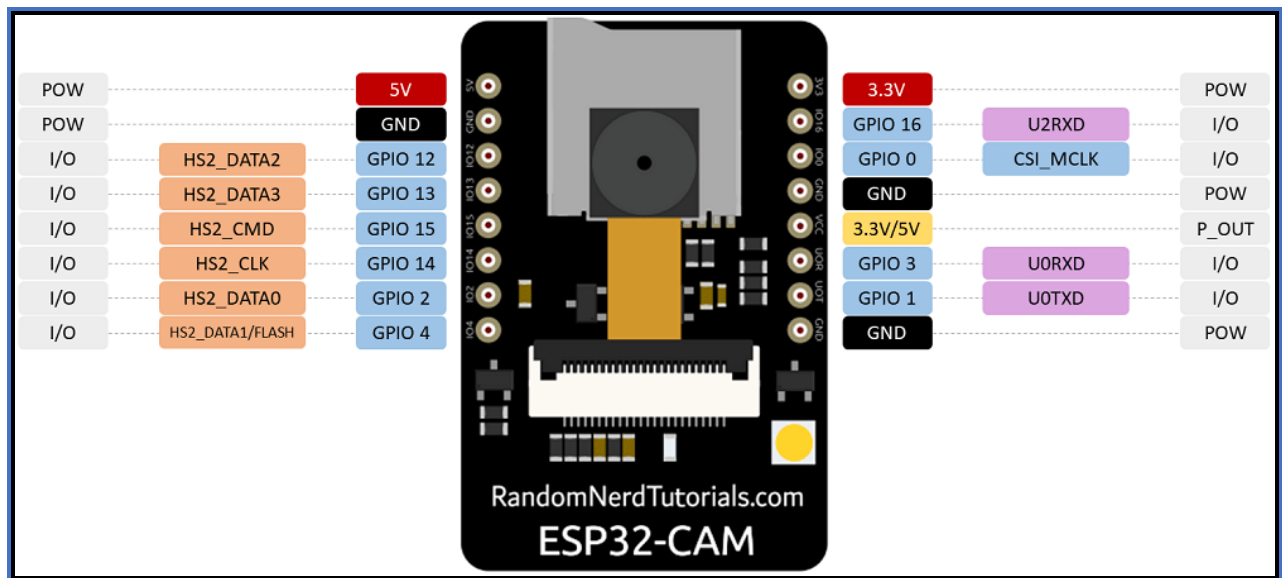
2. Implementação do projeto

a. Componentes

I. ESP32-CAM

Neste caso foi de grande serventia o uso da versão "ESP32-CAM", pelo seu tamanho, consumo baixo de energia, bom processamento e pelo fato de vir integrado uma câmera com bom processamento de imagem. Na imagem abaixo é mostrado sua frente e pinagens:

Figura 3: Desenho esquemático do ESP32



Fonte: randonnerdtutorials.com

I/O = entrada/saída;

pow = energia;

GPIO = entrada/saída (programável);

GND = terra;

HS2_... = fazem interface com o microSDCard e neste projeto não foram usados;

U0RXD = entrada de sinal serial (neste projeto foi usado para receber instruções e dados)

U0TXD = saída de sinal serial (neste projeto foi usado para enviar dados)

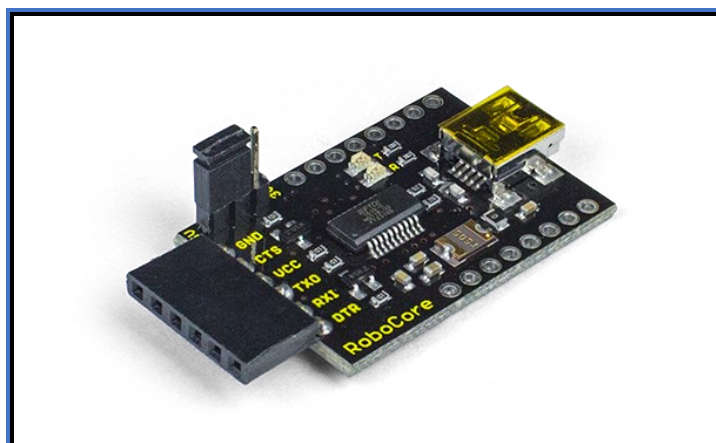
Informações Técnicas

- Processador: Xtensa® Dual-Core 32-bit LX6
- Memória Flash programável: 4 MB (dos quais 0,9 Mb são usados pelo *bootloader*)
- Memória RAM: 520 KBytes
- Memória ROM: 448 KBytes
- Clock máximo: 240 MHz
- Pinos Digitais GPIO: 11 (todos com PWM)
- Resolução do PWM: até 16 bits (ajustável via código)
- Wireless 802.11 b/g/n - 2.4GHz (antena integrada)
- Modos de operação: Access Point / Estação / Access Point + Estação
- Bluetooth Low Energy padrão 4.2 integrado

II. Módulo FTDI

Para fazer a interface entre as trocas de dados entre o ESP32-CAM e o computador foi usado uma placa FTDI "Future Technology Devices International" que converte a comunicação USB para Serial UART, sem a necessidade de portas extras no ESP32-CAM, bastando usar 2 jumpers um para a recepção de sinal na porta U0RXD do ESP32CAM, outro para o envio de sinal na porta U0TXD do ESP32CAM:

Figura 4: Placa FTDI usada



Fonte: www.robocore.net

III. Sensor cardíaco

O sensor de batimento cardíaco usado foi o descrito abaixo:

Figura 5: Sensor de batimento cardíaco



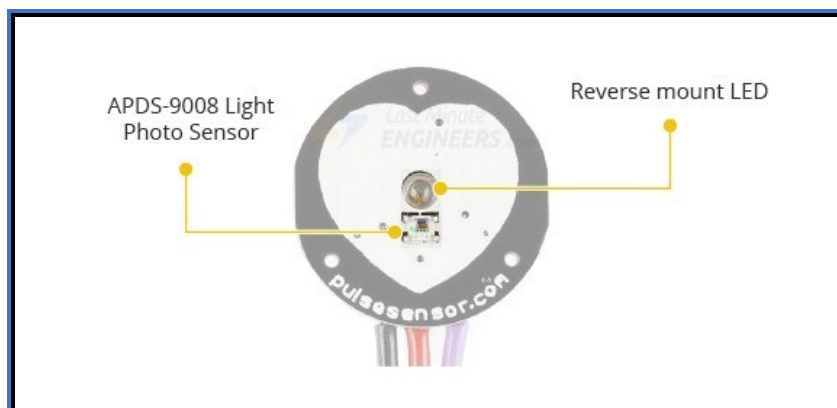
Fonte: robocore.com.br

Com 3 pinagens:

- + = entrada de 5v;
- = entrada de 0V;
- s = saída de sinal;

Na imagem abaixo temos uma foto da frente do aparelho, onde é encostado o dedo ou o pulso da pessoa a ser medida a frequência cardíaca:

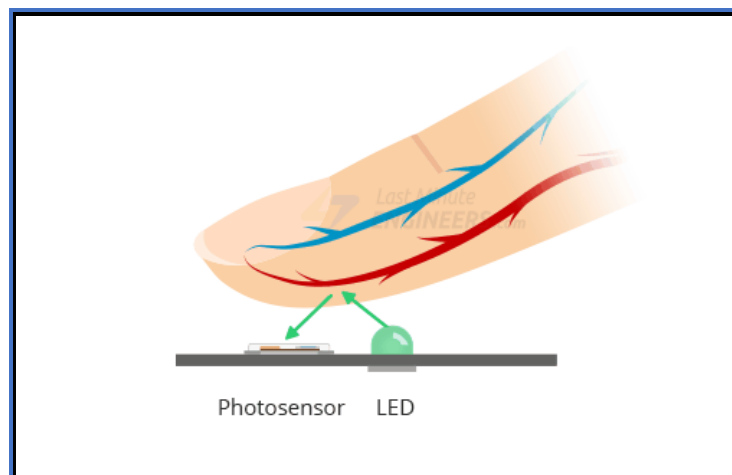
Figura 6: Frente do sensor



Fonte: lastminuteengineers.com

O princípio de funcionamento se baseia no método de leitura arterial chamado “Fotopletismografia” muito usado por médicos na leitura de batimentos cardíacos, se tratando de método indolor e sem incisões.[13] Como mostrado na foto abaixo:

Figura 7: Reflexão da luz verde no sangue



Fonte: <https://lastminuteengineers.com>

A hemoglobina oxigenada do sangue tem a função de absorver luz verde [13], quando coração bate cria-se uma pulsação que chega aos dedos e oxigena as hemoglobinas da região que acabam sorvendo luz verde e, portanto, gerando uma variação na quantidade de luz verde refletida de volta para o Photosensor. Assim, é possível saber a frequência cardíaca do usuário.

IV. Sensor de som

Neste caso, foi usado o sensor de som da “robocore.com”, descrito na imagem:

Figura 8: Sensor de som

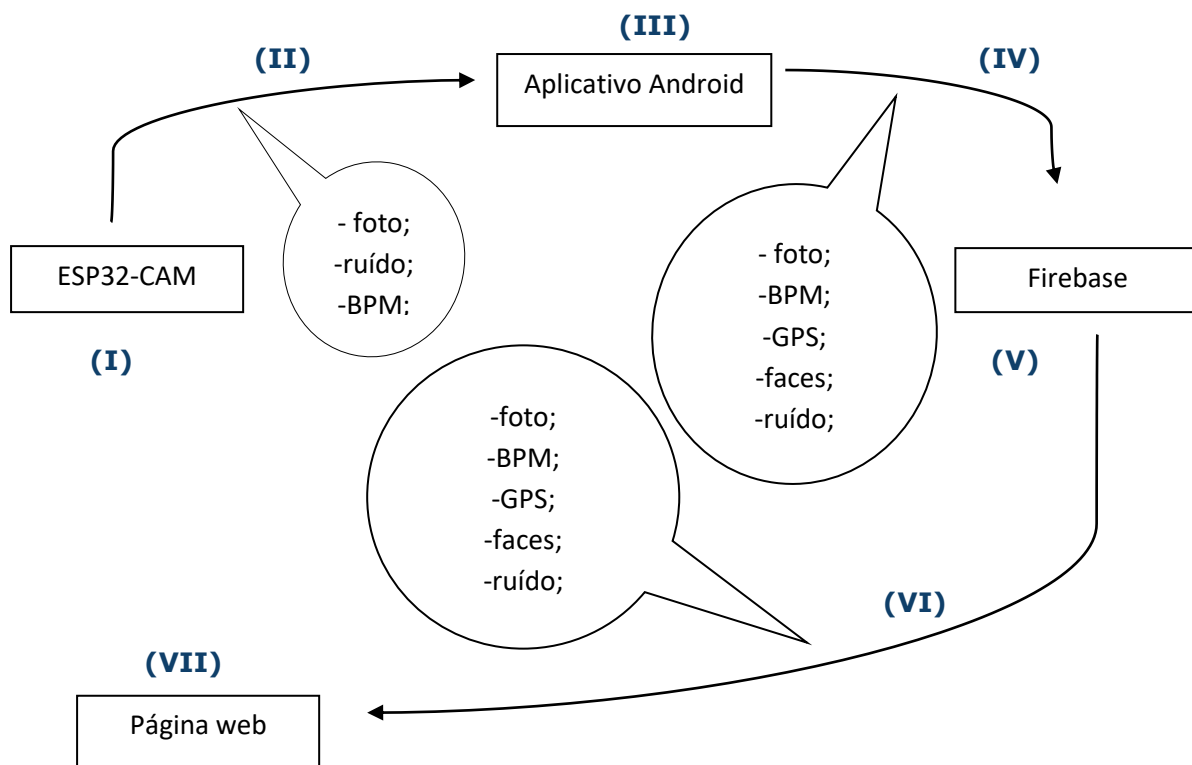


Fonte: www.robocore.com

Com tensão de entrada de 5V e saída de dados com amplificação de sinal com ganho de 100 para melhores resultados ao ser processado pelo Arduino ou ESP32 por exemplo.

b. Ciclo de dados do dispositivo

Abaixo segue ciclo de dados do projeto:

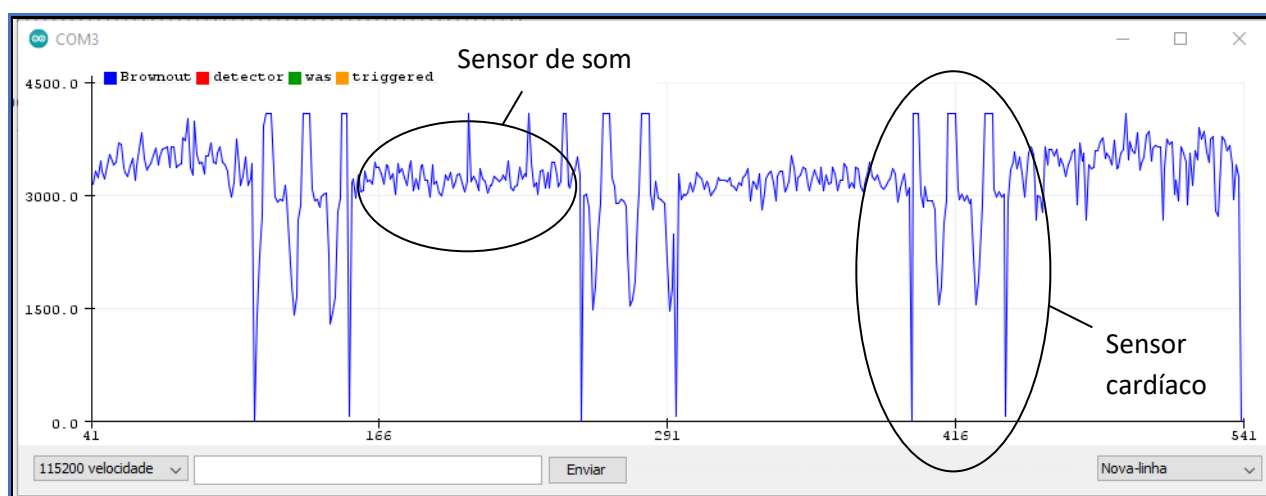


- (I) Acontece: captura de foto, captação de sinal do sensor de ruído e cálculo de nível de ruído médio, captação de sinal do sensor cardíaco e cálculo de batidas por minuto;
- (II) Os dados fluem do ESP32-CAM por Bluetooth de forma alternada (transmissão serial);
- (III) É exibido: nível de ruído a partir dos dados recebidos, foto processada para captura de faces com retângulo no rosto (se caso seja detectado), número de faces detectadas, batidas por minuto a partir do dado recebido e localização gps (usando o hardware do smartphone);
- (IV) Do Aplicativo Android, os dados fluem para o Firebase por autenticação e comandos de programação do aplicativo (linguagem Java);
- (V) Os dados são armazenados com identificadores de tempo atual para devida ordenação;
- (VI) Do Firebase os dados fluem por comandos de criação da página web (linguagem html5 e JavaScript);
- (VII) Acontece a exibição dos dados do Firebase com autenticação;

c. Gráfico do sensor de batimento cardíaco e sensor de som

O gráfico abaixo mostra dados obtidos pelo sensor de batimento cardíaco e pelo sensor de som:

Figura 9: Gráfico do batimento cardíaco + ruído do ambiente



Fonte: plotter do monitor serial do Arduino

São dois sinais de entrada, um do batimento cardíaco e o outro sensor de som, valores obtidos de forma reversada, pelo aplicativo de celular para o devido cálculo do BPM (batidas por minuto).

d. Cálculo do BPM

O cálculo aconteceu pelo próprio programa do Arduino, sendo feito com as medidas do início do primeiro pico de onda e com as medidas do segundo pico de onda. Conforme mostrado abaixo:

Trecho de Código do arduino:

$\text{deltaT} = \text{segundoBat} - \text{primeiroBat}$; (medidas do primeiro e segundo batimento em milissegundos)

```
// 1 batida ----- deltaT / (60 * 1000) minutos (conversão para minutos)
// x batidas ----- 1 minuto
// x = 60000/deltaT (cálculo do BPM)
```

$\text{batMin} = 60000 / \text{deltaT}$;

Medidas obtidas variando de 60 a 90 batidas/min, que está dentro da faixa de batimentos esperados[14].

e. Cálculo da qualidade do som

O cálculo da qualidade do som foi feito no Arduino e armazena todos os valores obtidos do sensor de som ao longo de 2 segundos (obtendo dados a cada 5 milissegundos) e faz a média. Foi usado o cálculo de potência média para dados de potência oscilando em um dado nó:

$$\text{Valor médio de Potência} = \frac{\sum_{i=0}^n (\text{dado } i - \text{nó})^2}{n} \quad (1)$$

n: número total de amostras de ruído;

dado i: amostra atual;

nó: o ponto zero do gráfico no qual oscila;

A partir desse resultado é classificado como ambiente calmo ou ruidoso.

f. Captura e envio da foto

Câmera digital: “dispositivo para fazer gravações digitais de imagens [5]. eles armazenam fotografias não como padrões de escuridão e luz, mas como longas cadeias de números. Isso tem muitas vantagens: nos dá fotografias instantâneas, nos permite editar nossas fotos e facilita o compartilhamento de fotografias usando celulares (celulares), e-mails e sites [6].”

Neste projeto foi usada uma câmera de 2MP (megapixels) código: OV2640. Foi testado o envio da foto por bluetooth e wifi, conforme descrito abaixo.

g. Envio de foto por wifi

Neste caso, foi criado através do servidor do roteador wifi residencial uma página web para armazenar as fotos capturadas pelo ESP32CAM. É gerado um endereço MAC no monitor serial do Arduino IDE (usado para fazer os programas do ESP32CAM) que passa a ser usado para acessar essas fotos. Porém, houve alguns problemas com relação a autenticação de site, pois o aplicativo Android não baixa imagens de sites sem Certificado Digital SSL, dificultando a viabilidade do método WIFI. Para que fosse viável o ideal seria hospedar a página em algum fornecedor, o que torna-se uma solução pouco versátil, principalmente para o usuário de tal dispositivo de segurança. Pensando nesse caso foi preferível o uso do Bluetooth conforme descrito a seguir:

h. Envio de foto por Bluetooth

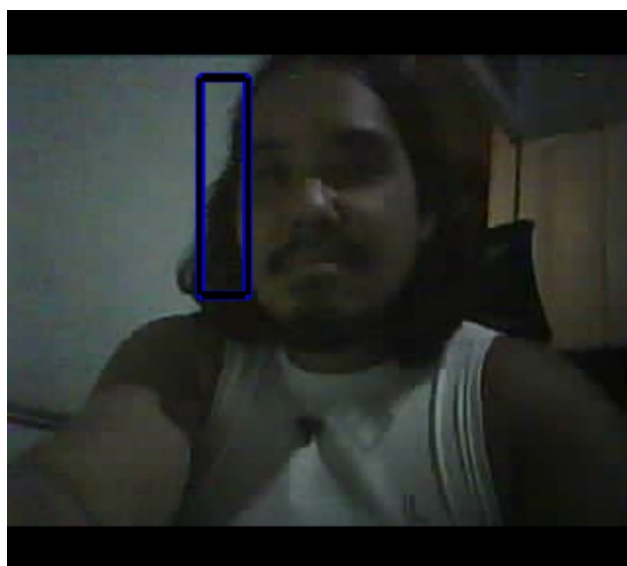
O Bluetooth® usado foi o 4.2 do próprio ESP32-CAM, usando o método Classic Bluetooth que consome mais energia, porém mais fácil de ser implementado. A obtenção das fotos foi feita por revezamento com o batimento cardíaco e o sinal de nível de ruído, pois a troca de informações entre o ESP32-CAM e o aplicativo Android é feita de forma serial. Neste caso foram usados três dados a foto, o batimento

cardíaco e o sinal de ruído de forma revesada. No aplicativo Android foi feito o processamento da foto com OPENCV para detecção de faces.

i. OpenCV

As fotos que chegam no aplicativo Android são processadas para se detectar faces, neste caso, foi usada a detecção por "haarscascades": arquivos ".xml" resultantes de aprendizagem por tentativa e erro com outras fotos com faces de seres humanos. Esses haarscascades exigem longo uso de aprendizagem para ficarem perfeitos, podendo chegar a meses de aprendizagem dependendo da máquina e do método de aprendizagem, muitas empresas não disponibilizam esses aprendizados devido a seu alto custo de produção. Segue abaixo amostra de foto processada e exibida no aplicativo:

Figura 10: Captura de imagem e processamento



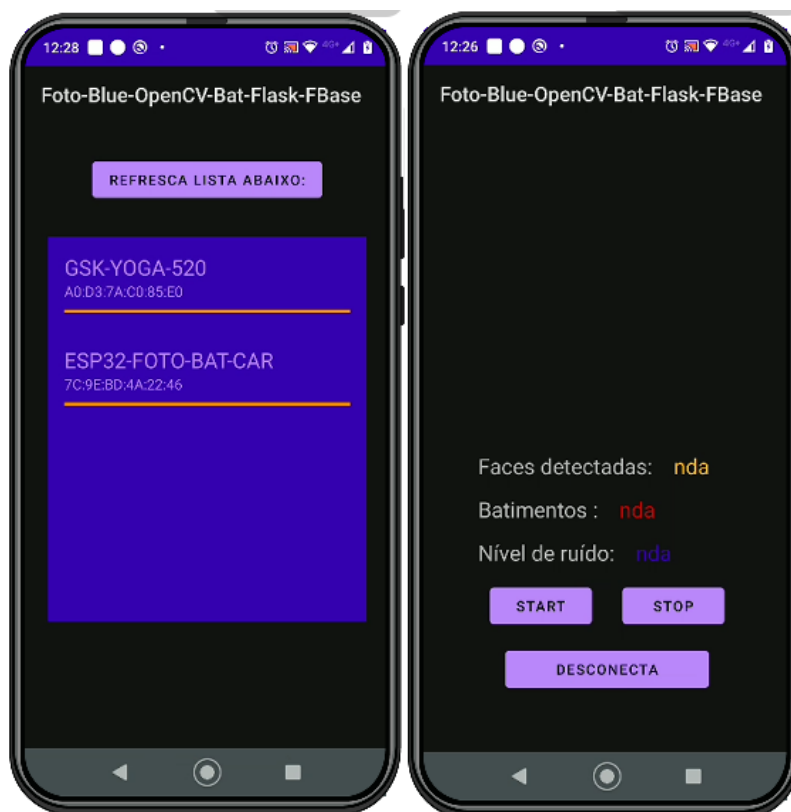
Fonte: Storage do Firebase

Como podem ver, a detecção não é perfeita (foi usada a detecção de faces do OPENCV já fornecido), sendo necessário melhor aprendizado para produzir melhores arquivos "haarscascades".

j. Aplicativo Android

O aplicativo foi desenvolvido do zero através de estudos pela internet e dicas ("stackoverflow.com" por exemplo), foi feito duas páginas interconectadas por botões. Inicialmente é feita a conexão com o Bluetooth abrindo uma próxima página que contém imagens capturadas pelo ESP32CAM, sendo feito o processamento de imagem, na tela é exibido o BPM do usuário, a quantidade de faces detectadas e o nível de ruído:

Figura 11: Página inicial e secundária do aplicativo



Fonte: produção do aplicativo no Android Studio

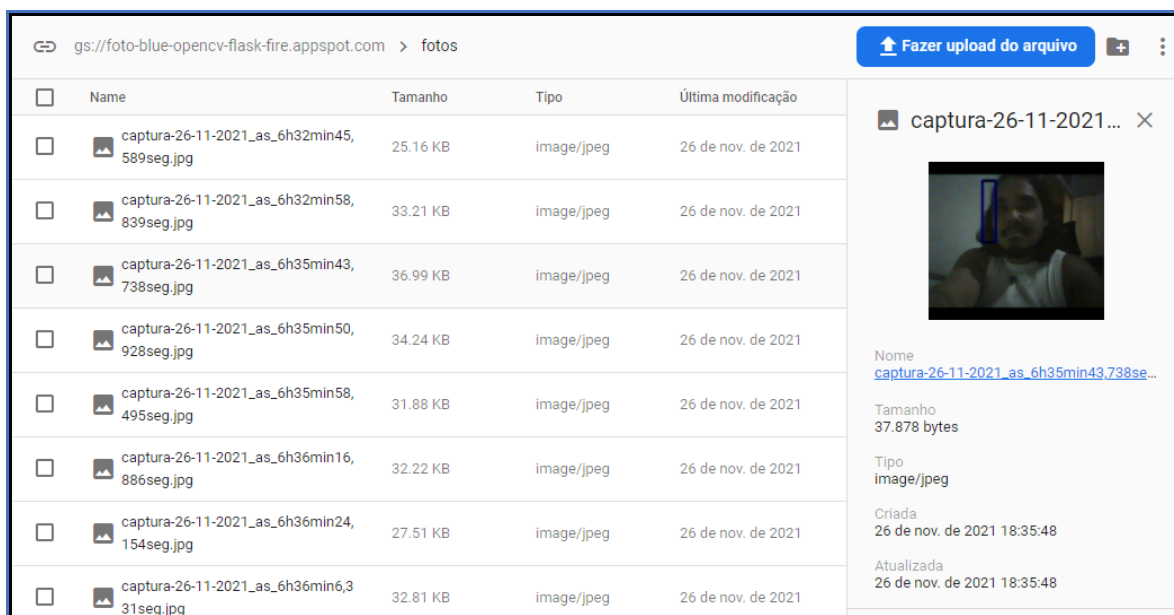
As fotos e o batimento cardíaco são enviadas a cada 5 segundos pelo ESP32CAM, tudo por Bluetooth.

k. Firebase e página web

Firebase: banco de dados pertencente ao Google localizados em lugares bem definidos ao redor do mundo, mediante locação paga ou gratuita.

A foto é enviada para o Storage (armazenagem de arquivos) do Firebase pela internet:

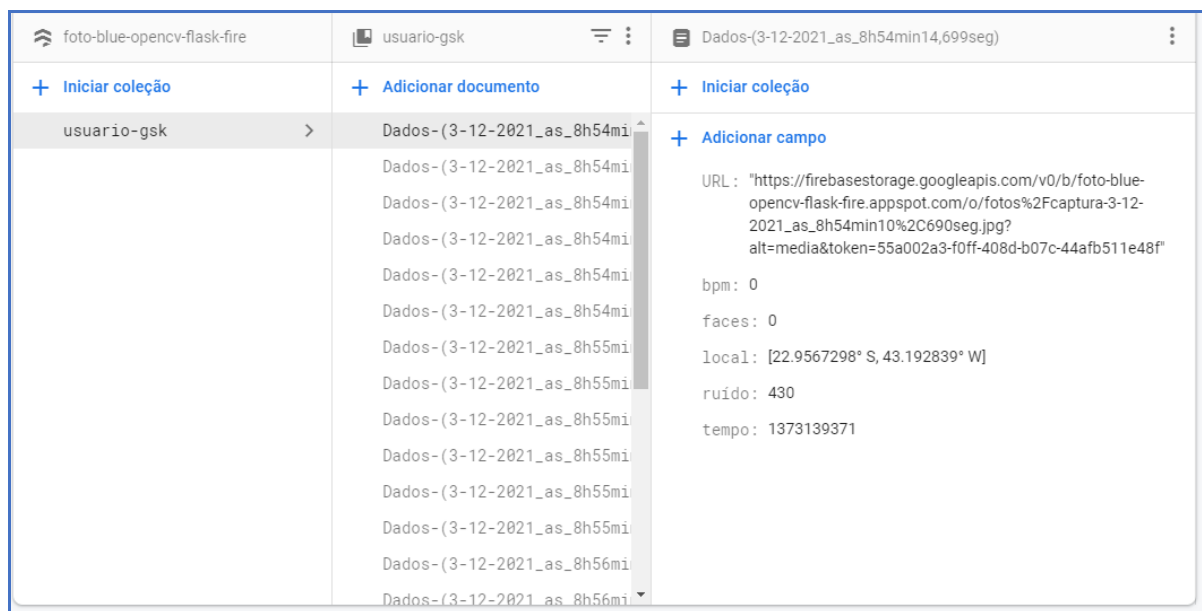
Figura 12: Storage do Firebase



Fonte: Storage do Firebase

Em seguida, é obtido seu url no próprio programa do aplicativo e, portanto, o batimento, a url, a localização gps, o ruído do smartphone e o tempo atual são enviadas pela rede wifi e armazenadas no Firestore do Firebase:

Figura 13: Firestore do Firebase

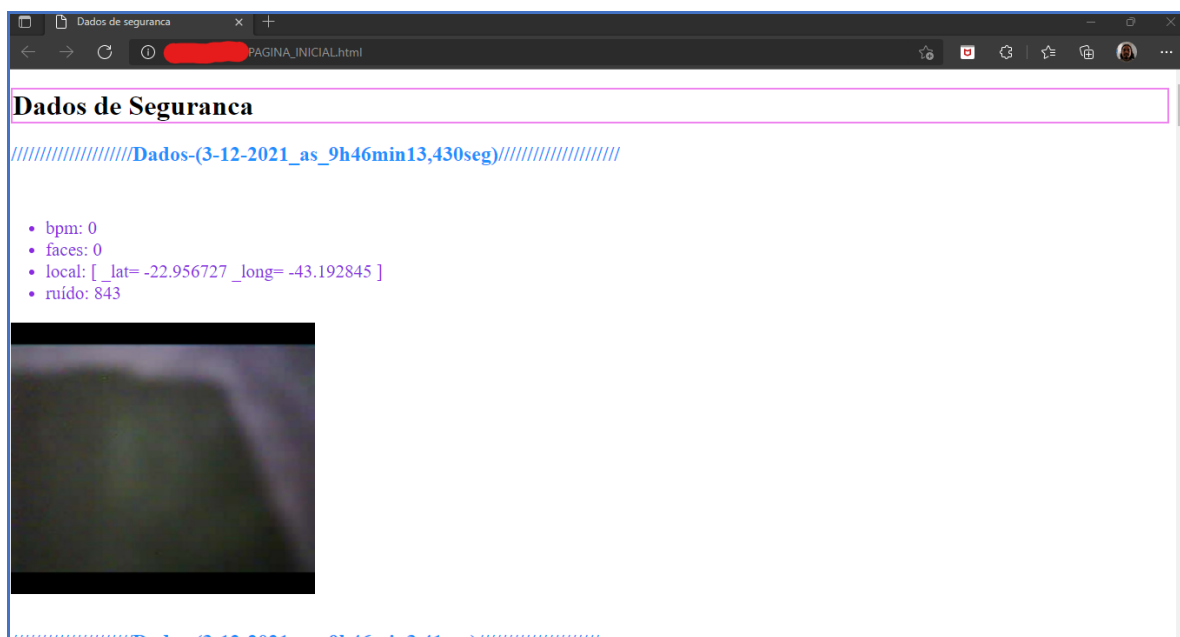


Fonte: FireStore em firebase.google.com

Obs.: Tudo com autenticação por e-mail e senha (neste caso, já embutidos no código do aplicativo).

E desse armazenamento é puxado os dados para uma página web, neste caso, foi usado o Visual Studio da Microsoft que possui um servidor próprio: "Live Server". Gerando-se essa página web exemplo:

Figura 14: Web page com os dados do Firebase



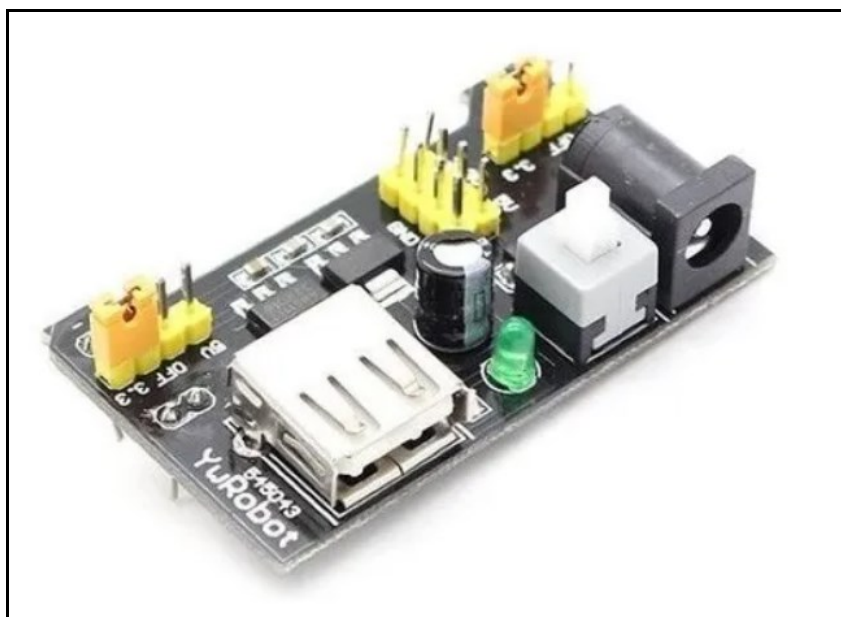
Fonte: servidor "live server" com o Visual Studio da Microsoft

Que possui todas as fotos e dados obtidos pelo ESP32CAM, pelo processamento de imagem do aplicativo Android e a localização GPS do smartphone.

I. Alimentação

Para alimentar o circuito foi usado uma bateria de 8 pilhas de 1,5v com o objetivo de proteger o circuito contra picos de correntes das fontes de tomada residenciais que poderiam comprometer o experimento. E um conversor de fonte dc para dc próprio para protoboards:

Figura 15: Fonte para protoboard



Fonte: www.mercadolivre.com.br

m. Ambientes de desenvolvimentos

Os ambientes de desenvolvimentos usados foram:

- Py Charm: para aprendizado do OpenCV;
- Arduino IDE: para aprendizado do sensor cardíaco e sensor de som, uso do ESP32(WIFI, Bluetooth);
- Android Studio: para o desenvolvimento do aplicativo Android;
- Visual Studio: para o desenvolvimento da página web;
- Firebase: banco de dados;

3. Conclusão e melhorias futuras

Neste projeto, foi obtido resultados satisfatórios no uso do aplicativo Android com seu ambiente de desenvolvimento Android Studio, o aprendizado da linguagem Java, JavaScript e HTML5 foi desafiador devido a problemas técnicos e de respostas complexas no uso do código, embora seu entendimento tenha sido relativamente fácil devido ao grande volume de informações disponíveis na internet. A montagem do ESP32-CAM com seus sensores se mostrou desafiador no início, principalmente no momento de comprar os equipamentos e, também, no uso da energia, neste caso por meio de pilhas. O Firebase se mostrou muito interessante e com bons recursos. A criação da página web foi desafiador com a escolha do ambiente de desenvolvimento neste caso o Visual Studio e na escolha de um servidor web neste caso foi escolhido o "Live server" já disponível no Visual Studio. O projeto funcionou muito bem e de forma muito responsiva com o uso do Bluetooth, consumindo pouca energia, mesmo com muitos sensores ao mesmo tempo. Os projetos no Arduino IDE (ambiente de desenvolvimento do ESP32) foram desafiadores no uso das bibliotecas e da linguagem c usada, embora exista grande volume de informações na internet.

No futuro, seria interessante desenvolver mais o produto a tal ponto de possivelmente haver sua fabricação e uso final. Os recursos são bem interessantes e certamente seria de grande serventia para situações de risco e envolvendo segurança. Seria interessante diminuir para algo fixo na roupa com câmeras frontal e laterais, com gps no próprio dispositivo, sensor cardíaco e sensor de som (possivelmente fazendo streaming do áudio direto no smartphone), envio dos sinais por bluetooth e com acesso ao banco de dados.

4. Bibliografia

- 1- GOOGLE - DEVELOPERS. **Visão geral do Bluetooth**. [S.l.]. Google, 2021. Disponível em: <https://developer.android.com/guide/topics/connectivity/bluetooth>. Acesso em: 1 set. 2021.
- 2- RODRIGO MARTINS. **Camera OV7670 + Arduino UNO + Programa + Software - português**. [S.l.]. YouTube, 2021. Disponível em: <https://www.youtube.com/watch?v=R6RoGqV17Vw>. Acesso em: 1 set. 2021.
- 3- GOOGLE - DEVELOPERS. **Android Studio**. [S.l.]. Google, 2021. Disponível em: <https://developer.android.com/studio?hl=pt-br>. Acesso em: 1 set. 2021.
- 4- ATIF PERVAIZ. **Bluetooth Kotlin - Android Studio Tutorial**. [S.l.]. YouTube, 2021. Disponível em: <https://www.youtube.com/watch?v=PtN6UTiu7yw>. Acesso em: 1 set. 2021.
- 5- DIGITAL camera. **Britannica Academic**. Disponível em: <https://academic-eb-britannica.ez370.periodicos.capes.gov.br/levels/collegiate/article/digital-camera/443006>. Acesso em: 7 ago. 2021.
- 6- WOODFORD, Chris. DIGITAL cameras. **Explain that stuff!**, 2020. Disponível em: <https://www.explainthatstuff.com/digitalcameras.html>. Acesso em: 7 ago. 2021.
- 7- **Arduino bluetooth camera (ABC)**. Project hub (Arduino). Disponível em: <https://create.arduino.cc/projecthub/alf81010/arduino-bluetooth-camera-abc-107c3d>. Acesso em: 7 ago. 2021.
- 8- THAT PROJECT. **ESP32 | BLUETOOTH CLASSIC | FLUTTER - *Realtime* Audio Recorder (ft. INMP441)**. [S.l.]. YouTube, 2021. Disponível em: <https://www.youtube.com/watch?v=AFGC9iYBxFQ>. Acesso em: 1 set. 2021.
- 9- 'Internet das Coisas': entenda o conceito e o que muda com a tecnologia. In: Globo. **TechTudo**. [S.l.]. 16 ago. 2014. Disponível em: <https://www.techtudo.com.br/noticias/2014/08/internet-das-coisas-entenda-o-conceito-e-o-que-muda-com-tecnologia.ghtml>. Acesso em: 5 nov. 2021.
- 10- BRASIL, Agência. Câmeras corporais diminuem letalidade em ações policiais em São Paulo. In: UOL - notícias. **Notícias**. [S.l.]. 13 jul. 2021. Disponível em: <https://noticias.uol.com.br/ultimas-noticias/agencia-brasil/2021/07/13/cameras-corporais-diminuem-letalidade-em-acoes-policiais-em-sao-paulo.htm>. Acesso em: 5 nov. 2021.
- 11- MOTOROLA NURSERY BR. **Review Completo Babá Eletrônica Motorola COMFORT85 CONNECT - Português BR**. [S.l.]. YouTube, 2021. Disponível em: <https://www.youtube.com/watch?v=4ngjQeEbG-w>. Acesso em: 5 nov. 2021.
- 12- TECH TERMS. **GPIO**. [S.l.]. TechTerms, 2014. Disponível em: <https://techterms.com/definition/gpio>. Acesso em: 5 nov. 2021.
- 13- DR. MARCO EDOARDO ARAUJO BEZERRA DE MELO. **Procedimentos e exames - Fotopletismografia**. [S.l.]. Vascular Clínica, 2019. Disponível em: <https://vascularclinica.com.br/procedimentos/fotopletismografia/>. Acesso em: 10 nov. 2021.
- 14- H3 MED. **Valores da frequência cardíaca normal, alta ou baixa**. [S.l.]. H3 Med, 2021. Disponível em: <https://www.h3med.com.br/blog/06/2020/valores-da-frequencia-cardiaca-normal-alta-ou-baixa/>. Acesso em: 21 nov. 2021.
- 15- OLIVEIRA, Sérgio de. **Internet das coisas com ESP8266, Arduino e Raspberry Pi**. São Paulo, SP: Novatec, 2021. color. ISBN: 978-65-86057-36-2. Disponível em: https://play.google.com/books/reader?id=rnojEAAQBAJ&pg=GBS.PP1.w.1.0.0_31&hl=pt. Acesso em: 29 nov. 2021.

- 16- **ESP32: technical reference manual.** Expressif Systems, 2021. Version 4,6. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf. Acesso em: 29 nov. 2021.
- 17- GOOGLE - DEVELOPERS. **Configurações de segurança de rede.** [S.l.]. Google, 2021. Disponível em: <https://developer.android.com/training/articles/security-config#:~:text=%20Network%20security%20configuration%20%201%20Add%20a,over%20HTTPS%2C%20you%20may%20want%20to...%20More%20>. Acesso em: 1 set. 2021.
- 18- STACKOVERFLOW. **Android 8: Cleartext HTTP traffic not permitted.** [S.l.]. StackOverFlow, 2021. Disponível em: <https://stackoverflow.com/questions/45940861/android-8-cleartext-http-traffic-not-permitted>. Acesso em: 1 set. 2021.
- 19- AWESOME DEV NOTES. **How to Allow insecure "http" url connection in an Android App using NetworkSecurityConfig?.** [S.l.]. YouTube, 2021. Disponível em: <https://www.youtube.com/watch?v=BhpixJmpPxE>. Acesso em: 1 set. 2021.
- 20- GOOGLE - DEVELOPERS. **Visão geral dos processos e threads .** [S.l.]. Google, 2021. Disponível em: <https://developer.android.com/guide/components/processes-and-threads>. Acesso em: 1 set. 2021.
- 21- STACKOVERFLOW. **how to set image from url for imageView.** [S.l.]. StackOverflow, 2021. Disponível em: <https://stackoverflow.com/questions/18953632/how-to-set-image-from-url-for-imageview>. Acesso em: 1 set. 2021.
- 22- GOOGLE - DEVELOPERS. **Button.** [S.l.]. Google, 2021. Disponível em: <https://developer.android.com/reference/android/widget/Button>. Acesso em: 1 set. 2021.
- 23- ASCII TABLE. **Ascii Table.** [S.l.]. Ascii Table, 2021. Disponível em: <https://www.asciitable.com/>. Acesso em: 1 set. 2021.
- 24- ARDUINO. **Serial.print().** [S.l.]. Arduino, 2021. Disponível em: <https://www.arduino.cc/reference/en/language/functions/communication/serial/print/>. Acesso em: 1 set. 2021.
- 25- SARTHI TECHNOLOGY. **014 : Bluetooth Image Transfer : Android studio bluetooth communication.** [S.l.]. YouTube, 2021. Disponível em: <https://www.youtube.com/watch?v=EzhWmZjEkrw&t=606s>. Acesso em: 1 set. 2021.
- 26- GOOGLE - DEVELOPERS. **ImageView.** [S.l.]. Google, 2021. Disponível em: <https://developer.android.com/reference/android/widget/ImageView>. Acesso em: 1 set. 2021.
- 27- STACKOVERFLOW. **Android set bitmap to Imageview.** [S.l.]. StackOverFlow, 2021. Disponível em: <https://stackoverflow.com/questions/15323844/android-set-bitmap-to-imageview>. Acesso em: 1 set. 2021.
- 28- STACKOVERFLOW. **How to get current date and time in Android?.** [S.l.]. StackOverFlow, 2021. Disponível em: <https://stackoverflow.com/questions/26880063/how-to-get-current-date-and-time-in-android>. Acesso em: 1 set. 2021.
- 29- IMASTERS. **Como utilizar o Parse GeoPoint em seu aplicativo Android?.** [S.l.]. Imasters, 2017. Disponível em: <https://imasters.com.br/android/como-utilizar-o-parse-geopoint-em-seu-aplicativo-android>. Acesso em: 1 set. 2021.
- 30- GOOGLE - DEVELOPERS. **Ver a última localização conhecida.** [S.l.]. Google, 2021. Disponível em: <https://developer.android.com/training/location/retrieve-current>. Acesso em: 1 set. 2021.
- 31- GOOGLE - DEVELOPERS. **Log.** [S.l.]. Google, 2021. Disponível em: <https://developer.android.com/reference/android/util/Log>. Acesso em: 1 set. 2021.

- 32- GOOGLE - FIREBASE. **Getting started with Firebase for the web – Firebase Fundamentals.** [S.l.]. YouTube, 2021. Disponível em: <https://www.youtube.com/watch?v=rQvOAnNvcNQ>. Acesso em: 1 nov. 2021.
- 33- GOOGLE - FIREBASE. **Primeiros passos com o Firebase Authentication em sites.** [S.l.]. Google, 2021. Disponível em: <https://firebase.google.com/docs/auth/web/start?hl=pt-br>. Acesso em: 1 nov. 2021.
- 34- GITHUB. **espressif/esp32-camera.** [S.l.]. Github, 2021. Disponível em: <https://github.com/espressif/esp32-camera>. Acesso em: 10 out. 2021.
- 35- APPTAURA. **Kotlin 101: How to communicate to a Bluetooth device Part 1.** [S.l.]. YouTube, 2021. Disponível em: https://www.youtube.com/watch?v=Oz4CBHrxMMs&list=PL_6gHICiRB_7JmbJsxayMnyQUQLhiYjb5&index=3&t=1998s. Acesso em: 10 out. 2021.
- 36- APPTAURA. **Kotlin 101: How to communicate to a Bluetooth device Part 2.** [S.l.]. YouTube, 2021. Disponível em: https://www.youtube.com/watch?v=eg-t_rhDSOM&list=PL_6gHICiRB_7JmbJsxayMnyQUQLhiYjb5&index=4. Acesso em: 10 out. 2021.
- 37- ANCODE. **Configure openCV in android studio arctic fox version 2020.3.1** [S.l.]. YouTube, 2021. Disponível em: https://www.youtube.com/watch?v=psoeNfFAKL8&list=PL_6gHICiRB_7JmbJsxayMnyQUQLhiYjb5&index=7. Acesso em: 10 out. 2021.
- 38- MIKE EXC. **Face Detection in Android using OpenCV and Java.** [S.l.]. YouTube, 2021. Disponível em: https://www.youtube.com/watch?v=GFGiQuEXKXY&list=PL_6gHICiRB_7JmbJsxayMnyQUQLhiYjb5&index=8. Acesso em: 10 out. 2021.
- 39- IA EXPERTS. **Deteccção de Objetos com Python.** [S.l.]. Udemy, 2021. Disponível em: <https://www.udemy.com/course/deteccao-de-objetos-com-python-e-opencv>. Acesso em: 10 out. 2021.
- 40- JAMILTON DAMASCENO. **Desenvolvimento web completo - 20 cursos + 20 projetos.** [S.l.]. Udemy, 2021. Disponível em: <https://www.udemy.com/course/web-completo>. Acesso em: 10 out. 2021.
- 41- JAMILTON DAMASCENO. **Desenvolvimento Android completo 2021 - Crie 18 Apps.** [S.l.]. Udemy, 2021. Disponível em: <https://www.udemy.com/course/curso-de-desenvolvimento-android-oreo>. Acesso em: 10 out. 2021.
- 42- GABRIEL TORRES. **Arquitetura de redes.** [S.l.]. Udemy, 2021. Disponível em: <https://www.udemy.com/course/redes-modulo-1>. Acesso em: 10 out. 2021.

