

# Sistema de Monitoramento de ECG Conectado por IOT

Gabriel Martins Ribeiro, 140020667  
Engenharia Eletrônica  
UnB - Faculdade do Gama (FGA)  
Email: gabrielmartinsgmr.10@gmail.com

Fabricio de Matos Alves, 202028196  
Engenharia Eletrônica  
UnB - Faculdade do Gama (FGA)  
Email: fabriciomatos02@gmail.com

## I. INTRODUÇÃO

O monitoramento de pacientes que possuem diagnóstico de problemas cardíacos ou têm a suspeita de desenvolvê-los, precisa ser feito de forma recorrente. Esse sistema busca minimizar esse problema através do monitoramento do sinal de ECG com sensores conectados na pele dos pacientes, podendo ser usado na casa do paciente a fim de evitar o contato social do paciente em um hospital.

Segundo a Organização Mundial de Saúde (OMS) as doenças cardiovasculares (DVC) são a principal causa de mortes no mundo, com aproximadamente 17,9 milhões de vidas perdidas no ano de 2019, número este que representa cerca de 32% de todas as mortes registradas no mundo. Ainda segundo a Organização, cerca de 75% das mortes causadas por DVCs ocorrem em países pobres ou em desenvolvimento devido a falta de assistência médica primária para a detecção dessas doenças ainda em estágios iniciais, sendo as populações mais pobres as mais afetadas. No Brasil, segundo o Ministério da Saúde, cerca 300 mil pessoas por ano são acometidas por casos de Infarto Agudo do Miocárdio, sendo destes, 30% levam a casos de morte, e é estimado que haverá aumento de casos de até 250% até 2040.

Segundo estudo realizado para o ano de 2015, os custos estimados relacionados a tratamento clínicos e tratamentos cirúrgicos de eventos de DVC na rede SUS é da ordem de R\$ 5.1 bilhões, além dos custos na ordem de R\$ 700 milhões com órteses, próteses e materiais especiais.

Como forma de classificar doenças e problemas relacionados a saúde, a OMS elaborou a CID (Classificação Estatística Internacional de Doenças e Problemas Relacionados com a Saúde) que em sua décima versão capítulo IX CID-10:IX agrupa as chamadas "Doenças do aparelho circulatório" que são codificadas com a letra I seguida de dois números. Doenças como Doenças Reumáticas de Valva (I05, I06 e I07), Doença Cardíaca Hipertensiva (I11), Infarto Agudo do Miocárdio (I22), Pericardite Aguda (I30), Hipotensão (I95), entre outras. Pacientes com tais doenças podem se beneficiar de um monitoramento eficaz e à distância de ECG junto do acompanhamento de médico especialista, a fim de obter tratamento adequado com antecedência.

## II. FUNDAMENTAÇÃO TEÓRICA

Eletrocardiografia (ECG) é o processo de aquisição de atividade elétrica do coração por um período de tempo usando eletrodos posicionados em contato com a pele do paciente. O ECG é considerado o melhor método para detecção de anormalidades cardíacas, onde os eletrodos detectam pequenas mudanças elétricas na pele provenientes dos padrões eletrofisiológicos de despolarização e repolarização durante cada batimento cardíaco.

Para o ECG de 12 canais, dez eletrodos são posicionados nos membros do paciente e na superfície do peito. A magnitude do potencial elétrico é medido em 12 ângulos diferentes e é gravado por um período de tempo (geralmente 10 segundos). Existem 3 componentes principais em um ECG : a onda P, que representa a despolarização do átrio; o complexo QRS, que representa a despolarização dos ventrículos; e intervalo QT, que representa a duração total da sístole (período de esvaziamento do ventrículo) elétrica ventricular. Além destes, outros pontos/ondas são observáveis em um registro de ECG, como:

- P representa o pulso de contração do átrio;
- Q representa a deflexão para baixo imediatamente antes da contração ventricular;
- R representa o pico da contração ventricular;
- S representa a deflexão para baixo imediatamente depois da contração ventricular;
- T representa a recuperação dos ventrículos;
- U representa o sucessor da onda T, mas é pequeno e quase nunca observado

Durante cada batimento cardíaco, um coração saudável tem uma progressão ordenada de despolarização que começa com "células marcapasso" no nó sinoatrial, se espalha pelo átrio, passa pelo nó sinoatrial até o pacote de HIS (anterior à bifurcação que leva aos ramos direito e esquerdo) e dentro das fibras de Purkinje, se espalhando para baixo e para a esquerda pelos ventrículos.

Na Figura 2 é apresentado o diagrama de blocos que representa o sistema do equipamento proposto de aquisição de ECG. A ideia principal do bloco Sensor de ECG é composto por sensores cardíacos posicionados na pele do paciente. O bloco Arduino é responsável pela aquisição e condicionamento

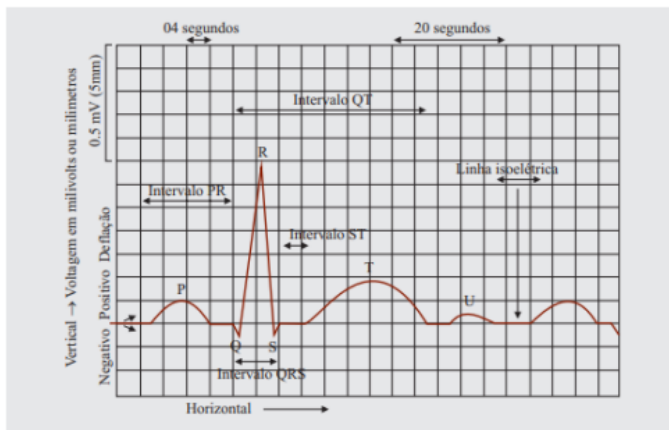


Figura 1: Caracterização do registro ECG. Fonte: ECG: manual prático do eletrocardiograma.

do sinal a partir do seu conversor A/D com resolução de 10 bits com clock variando podendo variar entre 50 kHz e 200 kHz para tal resolução. O bloco Raspberry Pi 3 modelo B é responsável por energizar o sistema de aquisição e apresentar o sinal registrado nos periféricos como computador, nuvem IOT e em um Website.

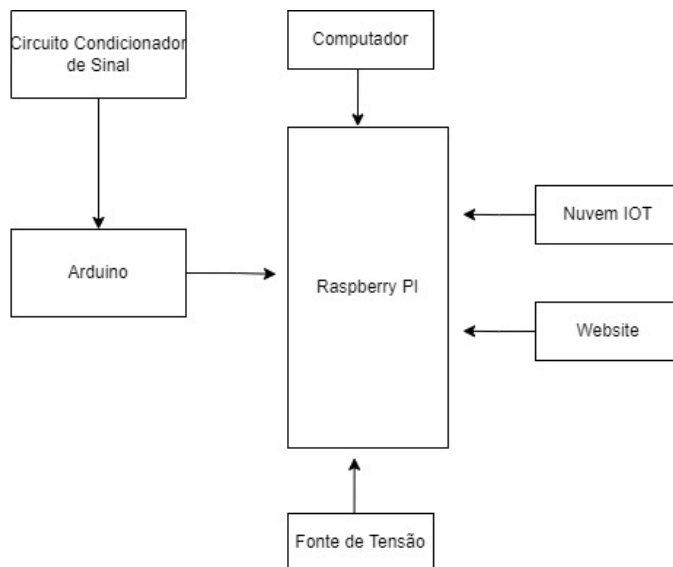


Figura 2: Diagrama geral proposto. Fonte: Autoria própria.

### III. PROTÓTIPO FUNCIONAL

- Esteremos usando o raspberry pi model 3B, pois atende os requisitos do nosso projeto, não terá problema de consumo de energia, pois o nosso sistema só funcionará conectado na rede elétrica. Usaremos um arduino UNO para obtenção e processamento dos sinais ECG. Além disso, o sistema não possuirá interface gráfica, pois as informações serão enviadas para outro servidor (clinica/medico).
- O sistema operacional utilizado será o Windows 10 Home com o software Ubuntu for Windows que é

suficiente para acessar os aplicativos e bibliotecas para o projeto, não sendo necessário a instalação de outro sistema operacional. O sistema da raspberry será o Raspberry OS (Raspbian), não será utilizado desktop, pois não terá interface gráfica.

- A instalação do sistema operacional foi feita, primeiramente, baixando o sistema no site da raspberry (<https://www.raspberrypi.com/software/operating-systems/>), foi instalado o sistema "Raspberry Pi OS with desktop and recommended software", após descompactar o arquivo baixado, foi usado um software que grava um arquivo de imagem de disco em um dispositivo removível (SD CARD) (<https://sourceforge.net/projects/win32diskimager/>). Depois, o cartão SD foi conectado a raspberry e conectando-a a energia, o sistema foi instalado automaticamente.
- A Raspberry foi conectada via cabo ethernet ao modem de internet para que, por meio do PC, pudesse obter o IP da raspberry, foi utilizado o seguinte software : <https://www.advanced-ip-scanner.com/br/>. A conexão foi feita por meio do software PUTTY (<https://www.putty.org/>) a partir do SSH. Nas configurações da raspberry o SSH e o VNC foram habilitados para acesso remoto via cabo ethernet e wi-fi no PC. No caso do projeto, a raspberry pi está conectada a rede local de internet e o acesso é via wi-fi.
- Quanto ao circuito condicionador de sinal, utilizou-se o módulo AD8232. Esse módulo adota um amplificador operacional usado para construir um filtro para eliminar o ruído. Ele implementa um filtro passa-alta de dois pólos para eliminar ruído por movimento do paciente. Conecta-se, também, 3 eletrodos na pele do paciente.
- A ferramenata de nuvem utilizada foi Ubidots

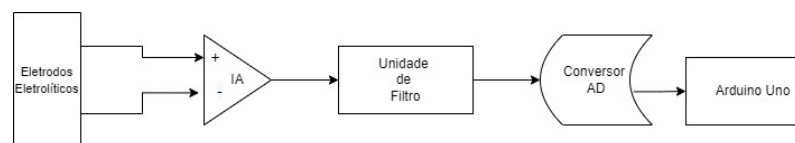


Figura 3: Circuito condicionador de sinal. Fonte: autoria própria

### IV. DESENVOLVIMENTO

Dentre os requisitos do sistema pode-se destacar:

- Baixo consumo de enrgia
- Alta acurácia
- Confiabilidade
- Tamanho pequeno

Os esquemático do projeto está mostrado na figura abaixo, está caracetrizando a conexão por meio da comunicação serial UART entre o arduino UNO e Raspberry PI, além disso, a

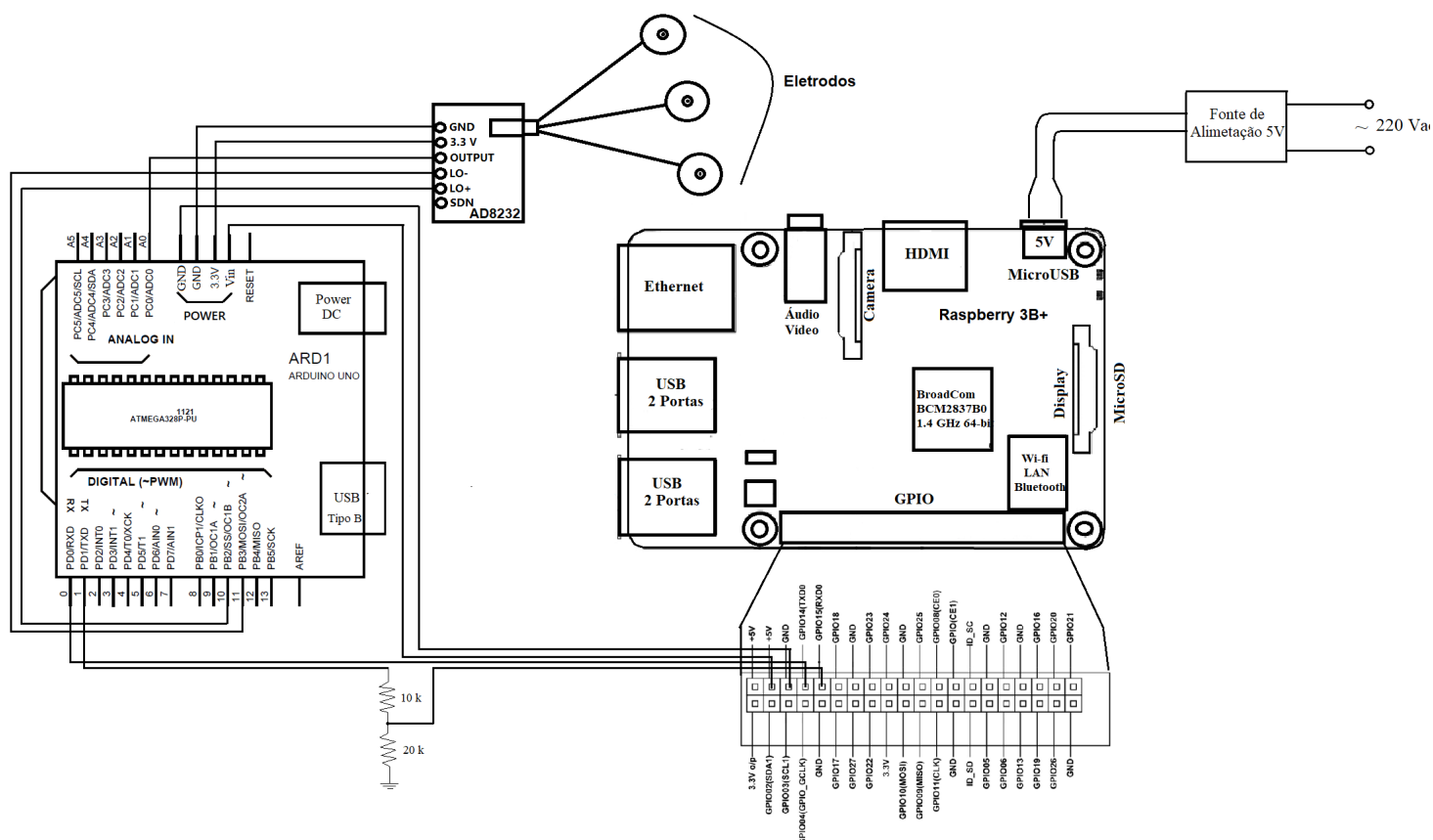


Figura 4: Esquemático do projeto com as conexões da Raspberry PI, arduino UNO e módulo AD8232. Fonte: autoria própria

conexão do módulo AD8232 que conecta os eletrodos com o arduino UNO que fará a leitura e transmissão dos dados.

Os dados lidos pela Raspberry PI serão enviados para uma aplicação IoT, gratuita e disponível, chamada Ubidots. A plataforma é uma API, onde os dados enviados pela raspberry PI se tornam visíveis através de gráficos atualizados em tempo real, sendo assim, possível visualizar o sinal ECG do paciente.



Figura 5: Interface inicial da plataforma Ubidots.

#### A. Testes de Aquisição do Sinal

A aplicação do sensor é feita conforme a figura 5, onde os eletrodos são distribuídos em posições estratégicas e definidas para uma boa captação do sinal. Quanto mais perto do coração estiverem as almofadas, melhor será a medição.

Os cabos são codificados por cores para ajudar a identificar o posicionamento correto.

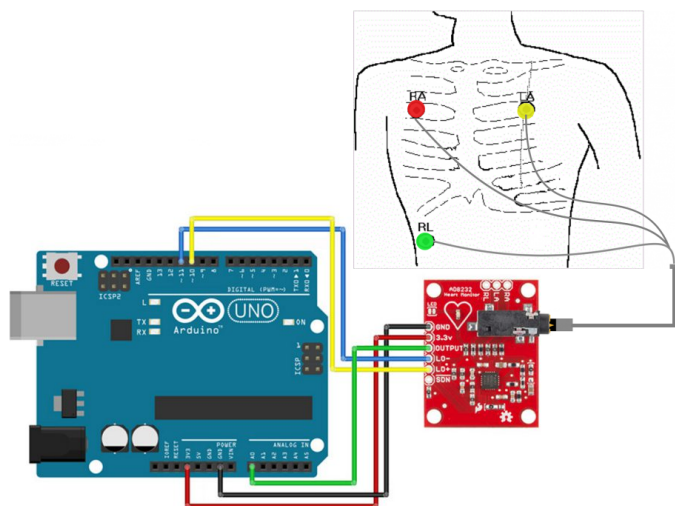


Figura 6: Sensor AD8232 e ligações dos eletrodos no corpo humano.

O código do arduino, mostrado em anexo, que faz a aquisição dos dados do módulo sensor AD8232 é estruturado da seguinte forma, a função setup configura porta serial para um

baud rate de 115200 bts, que é a velocidade de transmissão de dados, os pinos 2 e 3 do arduino são configurados para conectar aos Leads off positivo e negativo do módulo. Na função loop, é verificado se os pinos 2 e 3 estão em nível lógico alto, se sim, indica que o eletrodo está desligado, portanto, não há medição. Caso contrário, o valor lido é da porta analógica A0 é mostrado na tela.

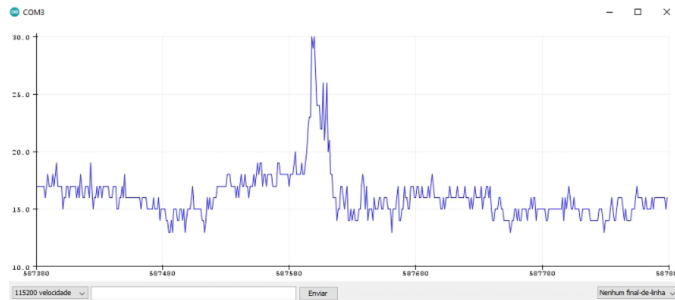


Figura 7: Signal obtido pelo plotter serial do arduino.

### B. Testes de Comunicação

A comunicação do arduino UNO com a raspberry PI foi feita através dos GPIO de transmissão e recepção dos dispositivos por meio da comunicação serial UART, onde foi realizado um teste para envio de dados do arduino para a raspberry. Na figura 4 mostra as conexões realizadas, o GPIO 14 (TX) da raspberry é conectado diretamente ao pino 0 (RX) do arduino, já o GPIO 15 (RX) é conectado por um divisor de tensão para 3,3 V que vem do pino 1 (TX) do arduino.

No código da raspberry Pi é utilizado a biblioteca *serial* com a função *read* para ler os bytes enviados do arduino, em seguida, utiliza-se uma função que converte os bytes em números inteiros, esses procedimentos ficam em loop infinito aguardando os dados que são enviados. Essa parte está em uma função chamada *receive\_data*. Há um segunda função chamada *post\_request* que faz uma requisição HTTP para o servidor da Ubidots, para isso teve que importar a biblioteca *requests*. Nessa função é criada a requisição HTTP, onde é necessário o token e nome da variável criada na plataforma Ubidots, essas informações são concedidas pela plataforma para serem incorporadas ao código e realizar a comunicação com sucesso para o envio dos dados em sequência. A função retorna *true* se a requisição for feita com sucesso e *false* se houve alguma falha de comunicação. E na rotina principal *main*, a função *receive\_data* é chamada com nome da variável configurada no site da Ubidots. O Ubidots recebe os dados do sinal ECG e monta um gráfico que pode ser visualizado na tela e que se atualiza a cada envio de dado.

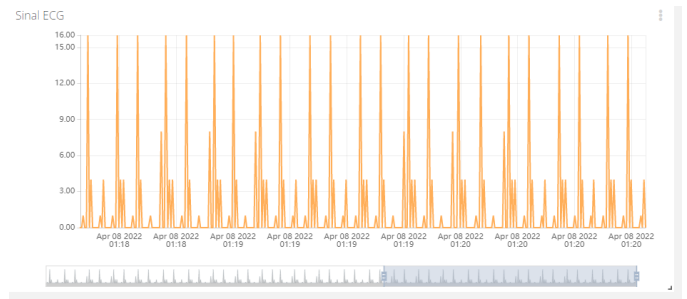


Figura 8: Gráfico do sinal ECG enviado da raspberry PI para o dashboard do Ubidots.

## V. BILL OF MATERIALS (BOM)

- Módulo AD8232
- 3 eletrodos eletrolíticos
- Raspberry Pi 3b
- Arduino Uno
- Laptop

## REFERÊNCIAS

- [1] Becchetti, C.; Neri, A.; Medical Instrument Design and Development: From Requirements to Market Placements; Ed 1; Chinchester: John Wiley & Sons, 2013.
- [2] Reis, H. J. L. (et al.); ECG: manual prático de eletrocardiograma; São Paulo: Atheneu, 2013. Disponível em: ECG-Manual-Prático-de-Eletrocardiograma-HCor.pdf (uff.br).
- [3] Siqueira, A. da S. E. (et al.); Análise do Impacto Econômico das Doenças Cardiovasculares nos Últimos Cinco Anos no Brasil; v.109; n.01; 2017. Disponível em: <https://doi.org/10.5935/abc.20170068>.
- [4] Lim, Y. G. (et al.); ECG Recording on a Bed During Sleep Without Direct Skin-Contact; IEEE Transactions on Biomedical Engineering; v.54; n.04; 2007.
- [5] <https://www.embarcados.com.br/arduino-taxa-de-amostragem-conversor-ad/#Leitura-padrao-do-AD-no-Arduino>.
- [6] [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)).
- [7] <https://www.medicinanet.com.br/cid10/i.htm>.
- [8] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR 60601-1: Equipamento eletromédico Parte 1: Requisitos gerais para segurança básica e desempenho essencial. Rio de Janeiro: ABNT, 2016.
- [9] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR 60601-1-2: Equipamento eletromédico Parte 1-2: Requisitos gerais para segurança básica e desempenho essencial — Norma Colateral: Perturbações eletromagnéticas — Requisitos e ensaios. Rio de Janeiro: ABNT, 2017.
- [10] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR 60601-1-6: Equipamento eletromédico Parte 1-6: Requisitos gerais para segurança básica e desempenho essencial — Norma colateral: Usabilidade. Rio de Janeiro: ABNT, 2011.
- [11] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR 60601-1-11: Equipamento eletromédico Parte 1-11: Requisitos gerais para segurança básica e desempenho essencial - Norma Colateral: Requisitos para equipamentos eletromédicos e sistemas eletromédicos utilizados em ambientes domésticos de cuidado à saúde. Rio de Janeiro: ABNT, 2015.
- [12] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR ISO 14971: Produtos para a saúde — Aplicação de gerenciamento de risco a produtos para a saúde. Rio de Janeiro: ABNT, 2009.
- [13] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT IEC/TR 80002-1:2020: Software de produto para saúde Parte 1: Orientação sobre a aplicação da ABNT NBR ISO 14971 a software para produtos para a saúde. Rio de Janeiro: ABNT, 2020.
- [14] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT IEC/TR 60601-4-2:2016: Equipamento eletromédico Parte 4-2: Orientações e interpretação — Imunidade eletromagnética: desempenho de equipamentos eletromédicos e sistemas eletromédicos. Rio de Janeiro: ABNT, 2016.

## VI. ANEXOS

### Código Python

#### Raspberry recebe os dados de ECG do Arduino

```

1  #!/usr/bin/env python3
2  import serial
3  import string
4  import time
5  import requests
6  import math
7  import random
8  import threading
9
10 TOKEN = "BBFF-L48Awxo7p33L5GSmURXCcaqt3CI5B1" # Put
    your TOKEN here
11 DEVICE_LABEL = "ECG_signal_monitoring_raspberry" #
    Put your device label here
12 VARIABLE_LABEL_1 = "ecg_sensor_data" # Put your
    first variable label here
13 #VARIABLE_LABEL_2 = "humidity" # Put your second
    variable label here
14 #VARIABLE_LABEL_3 = "position" # Put your second
    variable label here
15 #global payload
16 def receive_data(variable_1):
17     #abertura da porta serial
18     ser = serial.Serial('/dev/ttyS0', 115200,
19                         timeout=0)
20     ser.reset_input_buffer()
21     while True:
22         serialdata=ser.read()
23         serialdata= int.from_bytes(serialdata,"
24                                     little",signed=True)*(-1)
25         payload= {variable_1: serialdata}
26         post_request(payload)
27         #lendo dados da porta serial e printando
28         na tela
29     #return payload
30
31 def post_request(payload):
32     # Creates the headers for the HTTP requests
33     url = "http://industrial.api.ubidots.com"
34     url = "{}{}/api/v1.6/devices/{}".format(url ,
35                                             DEVICE_LABEL)
36     headers = {"X-Auth-Token": TOKEN, "Content-Type"
37               : "application/json"}
38
39     # Makes the HTTP requests
40     status = 400
41     attempts = 0
42     while status >= 400 and attempts <= 5:
43         req = requests.post(url=url, headers=headers
44                             , json=payload)
45         status = req.status_code
46         attempts += 1
47         #time.sleep(0.001)
48
49     # Processes results
50     print(req.status_code , req.json())
51     if status >= 400:
52         print("[ERROR] Could not send data after 5
53               attempts, please check \
54               your token credentials and internet
55               connection")
56         return False
57
58     print("[INFO] request made properly , your device
59           is updated")
60     return True
61
62 #def main():
63     #payload = receive_data(VARIABLE_LABEL_1)
64     #post_request(payload)
65
66 if __name__ == '__main__':

```

```

57 #while (True):
    #a = threading.Thread(target=post_request,name='
        Thread-a',daemon=True)
59 payload = receive_data(VARIABLE_LABEL_1)
    #b = threading.Thread(target=main,name='Thread-b
        ')
61 #a.start()
    #b.start()

```

## *Código Arduino*

### Arduino envia dados pra Raspberry

```

1 void setup() {
    Serial.begin(115200);
3    pinMode(3, INPUT); // Configura o para
        detec o de deriva es LO +
    pinMode(2, INPUT); // Configura o para
        detec o de leads off LO -
5 }

7 void loop() {

9     if((digitalRead(10) == 1)|| (digitalRead(11) == 1))
        {
            Serial.println('!');
11     } else {
        // envia o valor da entrada anal gica 0:
13     Serial.println(analogRead(A0));
    }
15     // Espere um pouco para evitar que os dados
        seriais saturem
    delay(1);
17 }

```