



**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE TECNOLOGIA**  
**FACULDADE DE COMPUTAÇÃO E TELECOMUNICAÇÕES**  
**ENGENHARIA DA COMPUTAÇÃO**

**Projetos de Engenharia III: Sistema de Monitoramento de Batimentos Cardíacos**

**Adriano Madureira dos Santos – 201706840041**  
**João Vitor Daltro Tavares Paiva -201606840076**  
**Victor Hugo Gonçalves Pantoja - 201706840003**  
**Wendell Williams Cardoso dos Santos - 201606840039**

**BELÉM-PA**

**2019**

## Sumário

1.	Introdução -----	3
2.	Desenvolvimento -----	3
2.1.	Objetivo -----	3
2.2.	Hardware e suas especificações -----	4
2.2.1.	Arduino -----	4
2.2.2.	PulseSensor -----	5
2.2.3.	ESP8266 -----	6
2.2.4.	GSM SIM800L-----	6
2.2.5.	Protoboard -----	7
2.2.6.	Jumpers -----	7
2.3.	Software e suas especificações -----	8
2.3.1.	Sketch Arduino -----	8
2.3.2.	Sketch ESP8266 -----	8
2.3.3.	Programa Servidor -----	9
2.4.	Orçamento-----	9
3.	Conclusão -----	9
4.	Referências Bibliográficas -----	10

## **1. Introdução**

A necessidade de tratamento das pessoas que possuem problemas cardíacos tem se tornado um fator agravante. Um dos motivos para que isso seja necessário é o fato de que os índices de mortalidade, relacionados à doenças cardiovasculares, tem crescido devido a uma série de fatores de risco. Dentre eles, podemos destacar: açúcar elevado no sangue, colesterol elevado, triglicéridos elevados, pressão arterial elevada, excesso de peso e obesidade, hábito de fumar, abuso de bebidas alcoólicas e pouco exercício físico.

Segundo a Organização Pan-Americana da Saúde (OPAS), as doenças cardiovasculares, como a taquicardia, são as principais causas de morte no mundo. Dessa forma, este trabalho tem por objetivo auxiliar pessoas durante o processo de tratamento de doenças cardíacas, permitindo também que os pacientes possam se locomover, não ficando apenas na área restrita do ambiente hospitalar. Isto implica também em uma possível diminuição da superlotação dos hospitais que tratam sobretudo de problemas cardiovasculares. Para isso, um sistema capaz de monitorar e alertar profissionais à respeito da situação atual dos pacientes foi desenvolvido. O sistema deve ser capaz de enviar dados referentes aos batimentos cardíacos, que são obtidos pelo sensor de frequência cardíaca, e também garantir que o profissional receberá um alerta, quando a frequência de batimentos cardíacos da pessoa estiver fora dos seus padrões.

## **2. Desenvolvimento**

### **2.1. Objetivo**

Dada as problemáticas quanto às doenças cardiovasculares, o alto custo de dispositivos de monitoramento cardíaco e o tamanho exacerbado de alguns destes, este projeto tem por intuito desenvolver um sistema em escala reduzida e de baixo custo de monitoramento dos batimentos cardíacos por minuto (B.P.M) em tempo real e que disponibilizará informações em um servidor, caso durante o monitoramento forem encontrados valores anormais que representem bradicardia (batimentos muito abaixo dos valores médios) e taquicardia (batimentos muito elevados dos valores médios) um alerta será enviado para os devidos profissionais.

Desta forma, este protótipo implementado utilizará ferramentas de baixo custo, como plataforma de desenvolvimento livre Arduino que integrará hardware e software servirá como controle para funcionamento dos componentes PulseSensor, sensor que fará a detecção dos batimentos, do ESP8266 e GSM SIM800L, módulos de comunicação que transferiram dados ao servidor. Logo, com o pleno funcionamento destes componentes, os dados obtidos pelo sensor serão disponibilizados em um servidor desenvolvido em linguagem de programação Python para o monitoramento do médico responsável pelo paciente.

## 2.2. Hardware e suas descrições

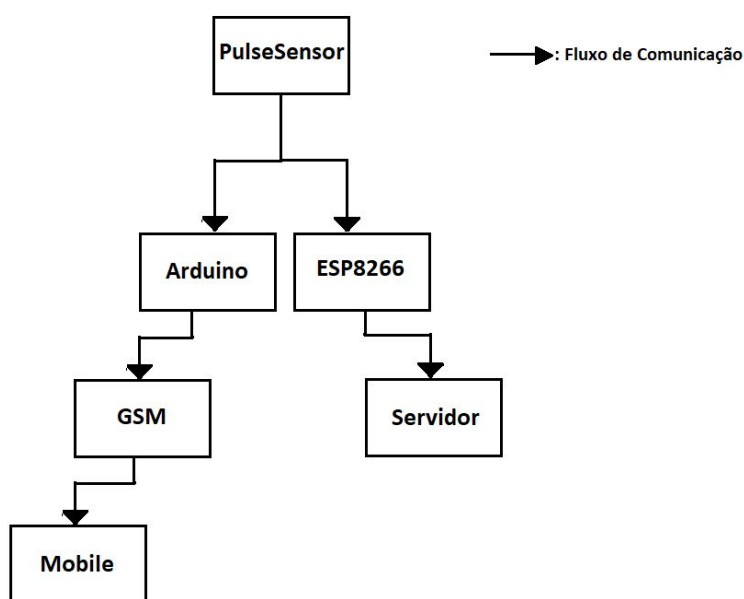


Figura 1 - Diagrama em alto nível do sistema.

### 2.2.1. Arduino

O Arduino é uma plataforma aberta e livre que permite o desenvolvimento de dispositivos eletrônicos capazes de mediar aspectos físicos (sensores), realizar ações (atuadores) ou se comunicar com outros dispositivos.

Suas especificações são:

- Alimentação: 9 a 12V;
- Tensão fornecida: 3,3 e 5V;

- 20 portas (14 digitais e 6 analógicas);
- EEPROM de 2 KB;

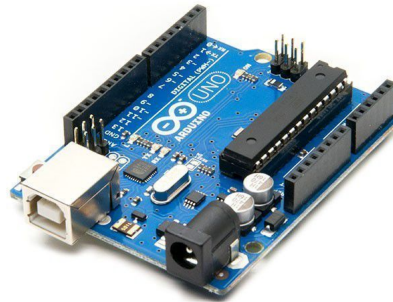


Figura 2 - Arduino.

### 2.2.2. PulseSensor

É um sensor de frequência cardíaca plug-and-play que combina um sensor óptico simples com amplificação e um circuito de cancelamento de ruído. Ele trabalha de maneira análoga a um exame de PPG (Fotopleletismografia) que calcula a frequência cardíaca através da variação dos níveis de oxigênio no sangue (SpO2). A luz é irradiada e através de sua reflexão nas artérias e veias é assim medida o SpO2 e consequentemente o B.P.M..

Suas especificações:

- Voltagem: 5V;
- Amperagem: 4mA;

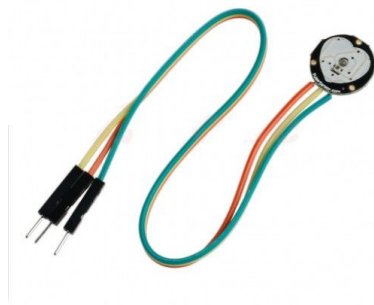


Figura 3 - PulseSensor.

### 2.2.3. ESP8266

O módulo Wifi ESP8266 NodeMCU é uma placa de desenvolvimento que combina o chip ESP8266, uma interface USB-serial e um regulador de tensão 3.3V. A programação pode ser feita usando LUA ou a IDE do Arduino, utilizando a comunicação USB. Isto é, é uma placa de desenvolvimento semelhante ao Arduino (possui algumas bibliotecas) que possui a tecnologia Wi-Fi.



Figura 4 - ESP8266 (NodeMCU).

### 2.2.4. GSM SIM800L

Este dispositivo foi utilizado durante o projeto para comunicar o alerta ao profissional, que é responsável por verificar o estado do paciente e tomar medidas de emergência. Possui diversas funcionalidades, dentre elas a capacidade de enviar mensagens SMS, realizar chamadas de voz e conectar-se a internet. Além disso, é conectado com o Arduino, que é o responsável por realizar a solicitação de alerta quando a frequência cardíaca do paciente estiver fora dos padrões.



Figura 5 - GSM SIM800L.

### **2.2.5. Protoboard**

Protoboard é uma placa com orifícios e conexões condutoras para montagem de circuitos elétricos experimentais. No projeto foi utilizada uma com 830 furos.

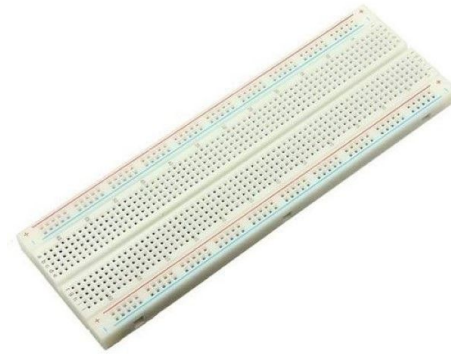


Figura 4 - Protoboard.

### **2.2.6. Jumpers**

Jumper é um pequeno condutor (fio metálico) utilizado para conectar dois pontos de um circuito



Figura 5 - Conjunto de jumpers.

## **2.3. Software e suas descrições**

Para a construção do sistema, foram necessárias duas sketches em linguagem C, desenvolvidas na IDE do arduino. Uma para controlar o ESP8266, e outra para o Arduino UNO.

O servidor UDP foi desenvolvido em Python, o qual também tem a capacidade de construir o gráfico a medida que vai recebendo dados inteiros.

### **2.3.1. Sketch Arduino**

Para este código foi necessário o código de exemplo da biblioteca do PulseSensor chamada “getting\_BPM\_to\_monitor” onde iremos captar o sinal emitido do sensor e com base nisso estimar as batidas por minuto (BPM) do usuário, pela modelagem:  $BPM = (\text{batimentos em } X \text{ segundos}) \cdot (60/X \text{ segundos})$

Implementamos também um contador para detectar anomalias nas BPMs, a fim de detectar uma constância nesta alteração, então quando o contador detectar um número de anomalias, que para este caso é 10. Este irá executar uma função definida por alertar via SMS algum responsável pelo usuário.

### **2.3.2. Sketch ESP8266**

Começamos importando as bibliotecas responsáveis por fazer as comunicações UDP e Wi-Fi. Criando um objeto da classe UDP, definimos a rede e senha ao qual o ESP irá se conectar. Este é um ponto crucial, a comunicação wireless para ser realizada com sucesso precisará ser configurada manualmente a cada mudança de rede, se houver, isto não afeta a mobilidade do sistema, visto que este tem objetivo de monitorar o paciente em uma área fixa.

Após a configuração do SSID da rede. Na etapa de setup, além de inicializar o pino D4 do ESP, executamos a função connect(), responsável por realizar a conexão à rede. Chegando ao loop do sketch a função send() será executada constantemente, enviando os dados lidos pela porta analógica A0 do ESP8266.

A função send() é quem monta o pacote UDP, primeiramente definindo o endereço de destinatário pelo comando “udp.beginPacket()” onde coloca-se o endereço ip do destino e a porta onde a escuta UDP está sendo realizada.

Feito isso, é realizado o comando de leitura da porta, que é gravado em uma variável e enviado via comando “udp.println()”.



### 2.3.3. Programa Servidor

O servidor ao ser iniciado, ficará escutando qualquer informação que chegar a porta a qual foi configurada, no caso a mesma porta definida no programa carregado pelo ESP. Construímos uma função para a montagem do gráfico, assim que o pacote chega, a informação é extraída e salva em um vetor para a plotagem, após 50 recebimentos a primeira informação recebida é apagada, para não sobrecarregar a memória, e assim por diante.

### 2.4. Orçamento

COMPONENTES	CUSTO
Arduino	R\$ 40,00
GSM SIM800L	R\$ 70,00
ESP8266	R\$ 40,00
Regulador LM2596	R\$ 10,00
PulseSensor	R\$ 18,00
Protoboard	R\$ 18,00
10 Jumpers	R\$ 4,00
<b>CUSTO TOTAL</b>	<b>R\$ 200,00</b>

Tabela 1 - Custo de projeto.

## 3. Conclusão

Os sistemas de monitoramento atuais se restringem ao ambiente hospitalar e frequentemente são focados acima de tudo para pacientes em estados mais críticos. Isso significa que o monitoramento de pessoas que possuem problemas cardíacos evidentes, entretanto menos frequentes, têm menor prioridade em relação ao atendimento. Dado este fato, o sistema é capaz de assegurar de uma maneira geral, que sua utilização é voltada a todos pacientes com os mais diversos estágios das doenças cardiovasculares, a fim de garantir a segurança desses pacientes pelo monitoramento.

A possível comunicação do Arduino com outros hardwares tais como ESP8266 e com GSM SIM800L, garante que os dados de seus batimentos cardíacos obtidos pelo sensor, sejam avaliados pelos profissionais de saúde, os quais podem tomar diversas medidas com base nos resultados de suas análises, como poder fazer realmente com que o paciente, que estava em estado de observação, retorne ao ambiente hospitalar por um possível agravamento de sua situação ou até mesmo em situações de emergência, em que se faz necessário a busca do paciente no local em que ele se encontra.

## 4. Referências Bibliográficas

- RIÇATO, Filipe T. *Proposta de um Sistema de Monitoramento em Tempo Real, para Estudos de Determinação de Economia de Energia, em Aquecimento Solar de Água*. 2018. 81 p. Trabalho de Conclusão de Curso - Universidade Estadual de Londrina, Londrina, 2018.
- BONIFÁCIO, Pedro H. Batista. *Desenvolvimento de Sistema Supervisório para Aquisição de Dados de Sensores Embarcados Através do Protocolo TCP/IP*. 2018. 98 p. Trabalho de Conclusão de Curso - Universidade Estadual de Londrina, Londrina, 2018.
- Cruz, L. F., Bernardon, C., Santos, E. V. M. S., Simões, W. C. S. S., & Silva, V. J. (2016). Um Sistema para monitoramento de sinais fisiológicos baseado em hardware de baixo custo com acesso via WEB. In *Simpósio Brasileiro de Redes de Computadores-Workshop de Comunicação de Sistemas Embarcados Críticos SBRC-WoCCES*.
- FORBES, Elliot. UDP Client and Server Tutorial in Python. Disponível em: <https://tutorialedge.net/python/udp-client-server-python/>. Acesso em: 21 de maio de 2019.
- PERES, Rita; SANTOS, António P. Cunha. Plotagem de dados “em tempo real” com python usando matplotlib. Disponível em: <https://www.revista-programar.info/artigos/plotagem-de-dados-em-tempo-real-com-python-usando-matplotlib/>. Acesso em: 21 de maio de 2019.
- WORLD FAMOUS ELECTRONICS LLC. Heartbeats in Your Project, Lickety-Split. [S. l.], 2018. Disponível em: <https://pulsesensor.com/>. Acesso em: 16 abr. 2019.
- BANZI, Massimo *et al.* Arduino. [S. l.], 2005. Disponível em: <https://www.arduino.cc/>. Acesso em: 16 abr. 2019.