

Universidade Federal do Rio Grande do Norte

Relatório Final: Sistema de Monitoramento Cardíaco

Ana Clara Nobre Mendes  
Gustavo Alves Bezerra

Natal - RN  
2017

## Sumario

1. Introdução
2. Descrição do projeto
3. Especificação
  - 3.1 Requisitos Funcionais
  - 3.2 Requisitos Não Funciona
4. Propostas Similares
  - 4.1 Exemplos
  - 4.2 Diferencial da proposta
5. Componentes do Sistema
  - 5.1 Componentes de Software
  - 5.2 Componentes de Hardware
6. Modelagem
  - 6.1 Diagrama de blocos
  - 6.2 Diagrama de classes
  - 6.3 Diagrama de sequência ou atividades
  - 6.4 Máquina de estados
7. Implementação
  - 7.1 Detalhes de implementação de hardware
  - 7.2 Detalhes de implementação de software
  - 7.3 Alterações nos requisitos ou especificação
  - 7.4 Dificuldades e soluções
  - 7.5 O que faltou ser feito
8. Conclusão e Trabalhos futuros
9. Referências

## Introdução

O desenvolvimento de vestíveis (wearables) tornaram-se comuns diante do avanço tecnológico do século. Seguindo essa cultura o nosso projeto propõe a utilização de um sensor de frequência cardíaca comunicando-se com um sistema de monitoramento acessível, utilizando-se de hardware de baixo custo e componentes miniaturizados para uso contínuo do usuário, permitindo monitorar o seu histórico de pressão mensal para fins não medicinais, apenas de orientação pessoal.

## Descrição do projeto

O projeto "Sensor de Frequência Cardíaca e Monitoramento" (SFC&M) propõe o desenvolvimento de um vestível de baixo custo e miniaturizado que permite ao usuário monitorar sua frequência cardíaca e pressão no dia a dia. Visamos desenvolver um sistema embarcado que mesmo que o usuário não saiba manipulá-lo inicialmente, poderá desenvolvê-lo e aprimorá-lo à sua necessidade a partir da documentação desenvolvida para ele.

Com licença de Código Aberto, garantindo que qualquer pessoa possa utilizar, modificar e compartilhar o código do projeto, permitimos que cada entusiasta dele possa personalizá-lo à sua necessidade, garantindo também possíveis melhorias e otimizações dos dados. Porém, o objetivo principal é que um usuário comum possa monitorar sua saúde com um software de fácil entendimento.

Visando facilitar tal entendimento, o sistema possui uma interface que exige pouca interação do usuário mas bastante intuitiva quanto ao que ele precisa monitorar, mostrando em tempo real os batimentos cardíacos, gráficos da pulsação e valores. Garantindo também que o usuário possa salvar seus dados em um arquivo de extensão .csv para leituras futuras e análises. Caso o aparelho não esteja conectado ao computador, também será possível para o usuário monitorar sua pressão. Para tal, foram utilizadas três LEDs, responsáveis por indicar se a pressão do usuário está saudável ou em estado de risco alto ou moderado. Caso só a interface gráfica não esteja disponível, não é possível para o usuário visualizar o valor da pressão.

## Especificação

### 1. Requisitos Funcionais

O projeto verifica a pressão e batimento cardíaco do usuário por meio de dois sensores oxímetros, além dos sensores utilizamos três *led's* 5mm para serem indicativos do estado da pressão do usuário de acordo com os valores obtidos. O software de monitoramento consiste no armazenamento dos valores obtidos pelos oxímetros, cálculo da média das saídas de cada um e interface gráfica de fácil entendimento ao usuário, com gráficos em tempo real e interação da frequência cardíaca via imagem.

Para transformar os dados obtidos por meio dos sensores em valores próximos a realidade, desenvolvemos funções de armazenamento e reconhecimento de frequência sistólica e diastólica cardíaca, criadas baseando-se em valores obtidos por meio da verificação de pressão de cinco voluntários com um equipamento médico de pressão tradicional. Com esses valores foi possível calcular a regressão linear de modo a convertê-los para os valores correspondentes de pressão.

A regressão linear é um cálculo da Estatística que consiste e, dada uma relação entre dois valores, achar a equação da reta que mais se aproxima. É possível, também, calcular a confiança dessa relação. Ou seja, uma porcentagem que define o quão a variável Y é dependente da variável X. Onde, no nosso caso, X é o valor de entrada dos sensores e Y é o valor da pressão medida com o equipamento médico. A fórmula da regressão linear é dada por:

$$Y = \beta_0 + \beta_1 X$$

Onde os valores de  $\beta_0$  e  $\beta_1$  são definidos por, respectivamente:

$$\beta_0 = \bar{Y} - \beta_1 \bar{X}$$

E

$$\beta_1 = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

Asdfa

Onde  $\bar{X}$  e  $\bar{Y}$  são, respectivamente a média dos valores X e Y ( $\bar{X} = \sum_{i=1}^n X_i / n$ ).

Ou seja, a média dos valores obtidos pelos sensores e a média das pressões.

Ao aplicar-se a fórmula da regressão linear nos valores obtidos, calculamos um coeficiente de correlação de 91%. Que é um valor excelente tendo em vista que apenas dois sensores foram utilizados para medição e poucos voluntários fizeram parte da pesquisa. O ideal seria um coeficiente de correlação acima dos 95% com uma pesquisa feita utilizando-se mais sensores e mais voluntários.

Para garantirmos a maior tolerância a falha desenvolvemos funções para que o software verifique se de fato o sensor está sendo tocado para as saídas dos valores não serem de fatores externos ou sofrerem influência, a led vermelha pisca para que o usuário saiba que sua pressão está sendo captada e nesse momento o sistema verifica se de fato está ocorrendo batimento cardíaco nesse momento, para então dar início ao sistema e monitorar os valores obtidos, gerando gráficos na interface.

## 2. Requisitos Não Funcionais

A miniaturização do projeto não foi executada, portanto, o nosso sistema é de difícil transporte, pois todos os componentes precisam ser transportados juntos, a alimentação de energia é dependente de um computador, pois é alimentado via USB e não por bateria. Impossibilitando o uso em qualquer lugar, os oxímetros utilizados são muito sensíveis, garantindo interferência dos fatores externos e portanto não garantindo a maior tolerância a falhas desejada. O projeto não tem autonomia energética e nem de processamento de imagem, o código do hardware pode ser compilado e dado *upload* no embarcado por memória interna ou externa, porém para gerar a interface foi necessário usar um software de processamento chamado “*Processing*” que compila o código da interface e processa os resultados do sistema embarcado via porta serial.

## Propostas Similares

### 1. Exemplos

[Veloguita<sup>1</sup>](#): Um dispositivo de orientação GPS e monitoramento de frequência cardíaca para ciclistas

Projeto [FilipeFlop<sup>2</sup>](#) para aprender a usar o sensor de frequência cardíaca no Arduino

### 2. Diferencial da proposta

O diferencial da nossa proposta é a integração do uso do hardware com o software de monitoramento de fácil compreensão. O nosso objetivo principal é a criação de um vestível acessível e que promova resultados mais próximos da realidade, para garantir isso, utilizamos dois sensores, para que assim seja retirada uma média dos valores das suas saídas, resultando em taxas mais assertivas das medições do usuário.

Além disso, todas as aplicações semelhantes encontradas apenas concentram-se em medir a pulsação do usuário (batimentos por minutos). Nossa proposta é utilizar os sensores de modo a inferir a pressão (sistólica e diastólica) do portador com razoável precisão.

## Componentes do Sistema

### 1. Componentes de Software

Software é um projeto disponibilizado de forma Open Source, com licença MIT do projeto “[2]World Famous Eletronics” que gera gráficos em tempo real do monitoramento de um sensor de frequência cardíaca, utilizando a linguagem de programação Processing, baseada em Java, que trabalha com arquivos, vídeos e imagens comunicando-se com projetos embarcados via porta serial.

O projeto cria formas de onda que representam a pulsação do usuário, executa um gif indicativo do batimento cardíaco para cada sensor, mostrando os valores captados por cada um e promove a interação do usuário via mouse e teclado. O primeiro serve para selecionar a porta serial na qual o embarcado está conectado, para que assim seja possível receber os dados captados e executados e o segundo para interatividade, em que o usuário poderá tirar uma cópia da tela do seu monitoramento, pela tecla “P”, para futuro uso, salvar a saída final da sua pulsação no momento que ele aperta a tecla “S” do teclado e resetar a captação dos dados ao apertar “R”. Essa integração ocorre com dependência de Software de terceiro, justamente porque é necessário rodar a aplicação no software de origem, chamado de Processing, como a linguagem utilizada, firmando a dependência de uma máquina para executar o projeto e não apenas o embarcado.

### 2. Componentes de Hardware

Arduino Uno

Sensores de frequência cardíaca (2 unidades)

Led verde 5mm

Led amarela 5mm

Led vermelha 5mm

Resistor 220R (3 unidades)

Protoboard

Jumpers macho-macho (6 unidades)

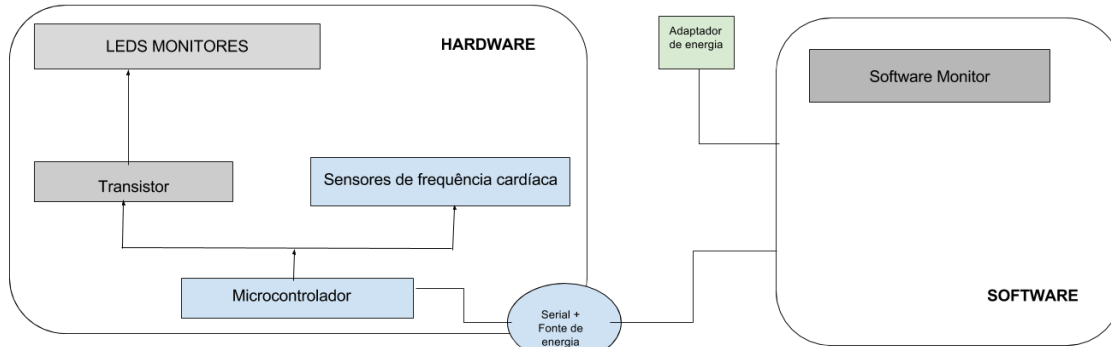
Jumpers macho-fêmea (6 unidades)

Ferro de solda de estanho

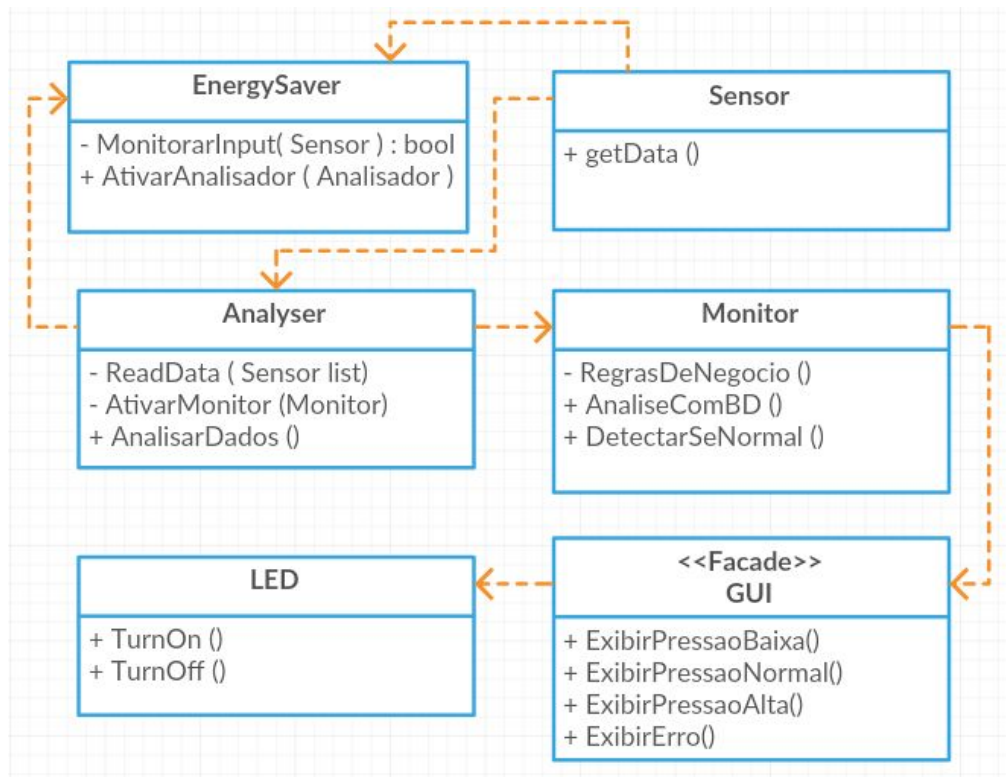


## Modelagem

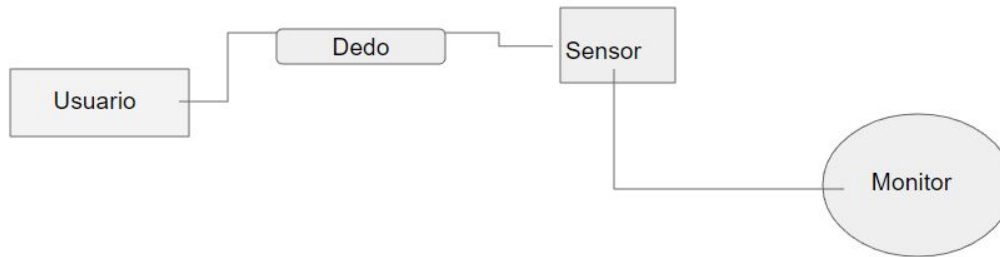
### a. Diagrama de Blocos



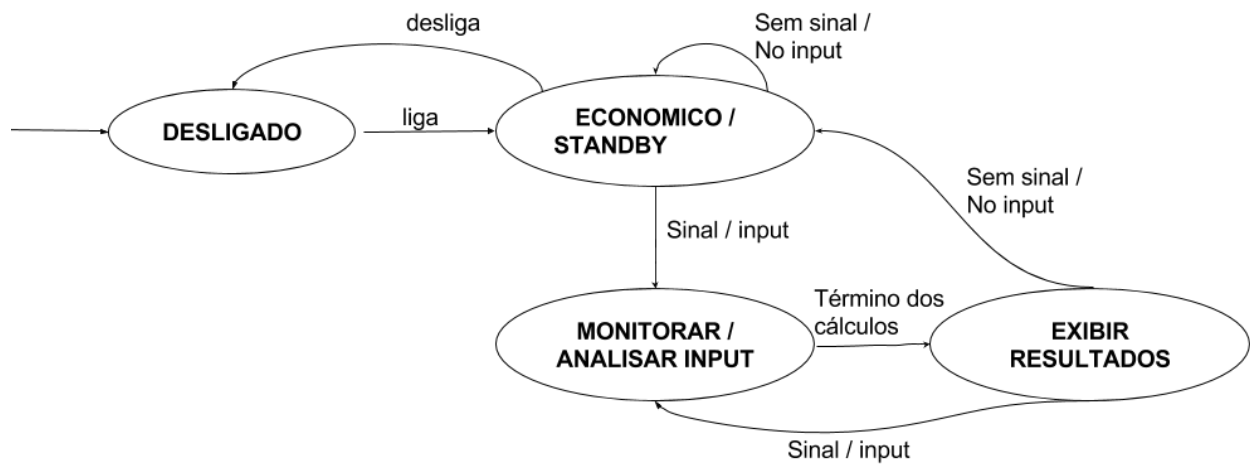
### b. Diagrama de Classes



c. Diagrama de Sequência ou Atividades



d. Máquina de Estados



## Implementação

### 7.1 Detalhes de implementação de hardware

O projeto contém dois sensores de frequência cardíaca combinados com um sensor óptico com amplificador cada, esses sensores enviam o sinal analógico para o microcontrolador, nós utilizamos o Arduino Uno para o desenvolvimento. O tamanho deles é bem reduzido(16 x 2,2mm), resultando em um baixo consumo de energia (cerca de 4mA) e a tensão de operação é de 3~5V. Para fins de monitoramento utilizamos três LED's de 5mm nas cores vermelho, amarelo e verde, que servem como indicadores luminosos para as saídas de verificação de pressão do usuário(vermelho para indicar que a pressão está sendo captada e saídas de pressão consideradas altas ou muito baixas, amarelo para pressões consideradas normais porém não necessariamente estáveis e verde para pressão saudável) e para cada LED utilizamos um resistor de 220R, que concluímos que seriam necessário resistências inferiores pois a luz das leds se tornaram insuficientes para o esperado. A comunicação desses componentes com o Arduino foi feita em uma protoboard para fins de desenvolvimento do hardware não miniaturizado e jumpers macho-machos foram utilizados para conexões de alimentação. Já os jumpers macho-fêmeas foram utilizados para os sensores se comunicarem com o microcontrolador e fonte de energia. Por fim, a comunicação com software foi desenvolvida via porta serial.

### 7.2 Detalhes de implementação de software

O que há de mais interessante a se relatar do software é o monitoramento. A interface gráfica apenas recebe os valores do monitoramento e os exibe na tela, além de salvar tais dados no arquivo quando requisitado. O código para exibição dos BPM utilizado está implementado e detalhado em [Pulse Sensor Amped - 2 Sensors](#) [2].

O algoritmo de monitoramento consiste, basicamente, nos seguintes passos:

1. Calcular a média dos valores lidos pelos sensores
2. Análise do gráfico cardíaco (máximo e mínimo local)
3. Computar se há ruído
4. Computar média das últimas medidas válidas
5. Aplicação da fórmula de regressão linear

À todo instante o algoritmo lê os dados advindos dos sensores. A média então é calculada e enviada para análise.

Na segunda etapa, a média recebida é utilizada para detectar se o valor recebido corresponde a algum máximo ou mínimo local. Caso esse seja o caso, o valor de entrada é um candidato a corresponder à pressão sistólica (máximo local) ou diastólica (mínimo local). Para se detectar o máximo e o mínimo local, armazena-se as duas últimas médias medidas pelos sensores. Sendo assim, caso a média anterior for

maior do que a média atual e a média mais antiga, ela (média anterior) será um máximo local. Uma lógica semelhante foi aplicada para se identificar os mínimos locais.

Na terceira etapa verifica-se o valor obtido da etapa anterior em relação às médias das pressões previamente computadas. Essa etapa faz-se necessária pois os sensores são bastante sensíveis e um ruído pode ter sido identificado como máximo ou mínimo local. Para evitar que esse ruído seja armazenado como valor válido de pressão, calcula-se o desvio padrão entre as pressões válidas previamente obtidas. Para o valor obtido ser considerado um valor válido de sistólica, ele tem de ser maior que o valor da diastólica mais o desvio padrão (um valor pouco maior que a média). Já para ser um valor válido de diastólica, necessita ser menor que a sistólica menos o desvio padrão (valor pouco menor que a média).

Os últimos dez valores válidos de cada pressão são armazenados e suas médias são consideradas o valor da pressão do usuário. Na quarta etapa, ao se identificar um valor válido de sistólica (provido da etapa anterior), substitui-se o valor mais antigo da pressão pelo recém-calculado e a média da pressão sistólica é calculada novamente.

Na última etapa, antes de se enviar os dados para o Processing salvá-los num arquivo, aplica-se a fórmula da regressão linear nos valores de modo a convertê-los para valores válidos de pressão.

### 7.3 Alterações nos requisitos ou especificação

Não houveram alterações.

### 7.4 Dificuldades e soluções

Uma das dificuldades encontradas foi em relação a um dos sensores. Tivemos um problema na solda, que teve de ser refeita. Por consequência, sua sensibilidade foi afetada: os valores lidos variavam menos em relação ao outro. Tal estabilidade, originalmente um problema, foi útil para que pudéssemos amenizar a sensibilidade exacerbada do outro sensor, obtendo, assim, resultados mais constantes.

Tivemos também dificuldades em relação à interface gráfica. Uma das nossas ideias iniciais seria que o usuário digitasse seu nome em um campo de formulário (*text box*) e, ao apertar um botão de salvar, um arquivo fosse criado com seu nome e sua pressão armazenada. Como não conseguimos implementar o botão, a solução encontrada foi de, ao apertar uma tecla específica, um único arquivo com nome pré-estabelecido pelos desenvolvedores fosse criado ou sobrescrito.

### 7.5 O que faltou ser feito

Em relação às funcionalidades não implementadas temos a miniaturização do protótipo. Essa etapa é fortemente influenciada pela implementação do algoritmo e

estabilidade do software e dos sensores. Logo, optamos por nos concentrar na estabilização, enquanto a miniaturização teve de ser postergada.

Além disso, conforme mencionado na seção anterior, não implementou-se a flexibilidade para salvar os arquivos. Ou seja, todas as medições de pressões são salvas em um único arquivo ao invés de ser criado um arquivo diferente para cada pessoa (um histórico distinto para cada).

## 8. Conclusão e Trabalhos futuros

Concluimos que, é viável desenvolver uma aplicação para medição de pressão utilizando apenas sensores, apesar de não haver nenhuma solução atualmente.

Para disponibilizar tal software, seria necessário, porém que o algoritmo desenvolvido fosse testado com vários sensores diferentes, e para valores mais precisos e aceitos por órgãos regularizadores de saúde seria necessário testes em mais voluntários, numa escala muito mais probabilística a falhas e um aferidor de pressão mais preciso. Sendo assim, possível de obter uma fórmula de regressão linear mais condizente com a realidade.

Para trabalhos futuros, faz-se necessário a miniaturização do protótipo e, conseqüentemente sua otimização (garantindo que funcione num hardware mais limitado). Será necessário, também, um teste de qualidade dos sensores; de modo a garantir que os valores exibidos pelo algoritmo não estão sendo influenciados por defeitos dos mesmos. Por último, é necessário revisar o código de modo a aprimorar a remoção de ruídos, certificando-se de que todos os valores utilizados para cálculo de pressão são válidos.

## 9. Referências

- [1] [https://github.com/WorldFamousElectronics/PulseSensor\\_Amped\\_Arduino](https://github.com/WorldFamousElectronics/PulseSensor_Amped_Arduino)
- [2] [https://github.com/WorldFamousElectronics/PulseSensorAmped\\_2\\_Sensors](https://github.com/WorldFamousElectronics/PulseSensorAmped_2_Sensors)
- [3] <https://www.filipeflop.com/blog/aprenda-usar-o-sensor-de-frequencia-cardiaca/>
- [4] <https://pulsesensor.com/pages/processing-visualization>
- [5] <https://www.filipeflop.com/produto/sensor-de-frequencia-cardiaca/>
- [6] <https://veloguia.wordpress.com/>
- [7] <http://blog.novaeletronica.com.br/circuito-monitor-de-batimentos-cardiacos/>
- [8] <http://www.discombobulate.me/workshop/arquetipos/pulse-sensor/>
- [9] <http://sistemaolimpo.org/midias/uploads/3f1fe0d1e16395d471bc613987158f4d.pdf>
- [10] Simulação feita no TikerCAD: <https://goo.gl/gjBG76>