



**CENTRO UNIVERSITÁRIO DE BRASÍLIA- UnICEUB**  
**PROGRAMA DE INICIAÇÃO CIENTÍFICA**

**CAIO MACEDO SANTOS**  
**LEONARDO YUNDI AIKAWA**

**DISPOSITIVO QUE AVALIA SINAIS BIOLÓGICOS PARA PREVER EPISÓDIOS  
HIPOGLICÊMICOS EM DIABÉTICOS TIPO 1 USANDO SENSORES DE UMIDADE,  
TEMPERATURA E VARIABILIDADE DA FREQUÊNCIA CARDÍACA CONTROLADOS  
POR PIC**

**BRASÍLIA**  
**2020**



**CAIO MACEDO SANTOS  
LEONARDO YUNDI AIKAWA**

**DISPOSITIVO QUE AVALIA SINAIS BIOLÓGICOS PARA PREVER EPISÓDIOS  
HIPOGLICÊMICOS EM DIABÉTICOS TIPO 1 USANDO SENSORES DE UMIDADE,  
TEMPERATURA E VARIABILIDADE DA FREQUÊNCIA CARDÍACA CONTROLADOS  
POR PIC**

Relatório final de pesquisa de Iniciação Científica apresentado à Assessoria de Pós-Graduação e Pesquisa.

Orientação: Prof. MSc Francisco Javier de Obaldía Díaz

**BRASÍLIA**

**2020**

## RESUMO

Tratamentos para disfunções como o Diabetes Mellitus, que traz consigo um descontrole glicêmico, trazem a cada dia inovações e melhorias, com o uso, por exemplo, do sensor de glicemia. O episódio hipoglicêmico é um dos sintomas que mais afeta diabéticos no mundo inteiro, levando a sintomas neuroglicopênicos como fome, tontura, fraqueza, dor de cabeça, confusão, convulsão e até mesmo coma, havendo grande necessidade de prevenção e controle dela. Contudo o uso de sensores glicêmicos para a previsão de hipoglicemias é de alto custo. Por ser um equipamento que é inserido sob a pele, pode causar irritação no local e grande desconforto para o seu usuário. Com a pesquisa realizada e tomando como base trabalhos já existentes na área, observou-se que existe a possibilidade de se estabelecer uma relação entre episódios hipoglicêmicos e sinais biológicos afetados pela manifestação do Sistema Simpático (sudorese, taquicardia, apreensão, tremor). Partindo deste princípio, propôs-se o desenvolvimento de um equipamento capaz de prever hipoglicemias noturnas por meio de um sensor de batimento cardíaco juntamente com um sensor de temperatura e umidade corpórea, para alertar o usuário prestes a ter um episódio hipoglicêmico. Este aparelho proposto teria seu diferencial pela forma não intrusiva de coleta de informação por meio dos sensores e com menor custo comparado ao CGM (Continuous Glucose Monitoring), atualmente disponível no mercado. Durante a realização desse projeto o Covid-19 encadeou uma pandemia mundial, levando à identificação do grupo de risco – do qual os diabéticos fazem parte – e isolamento social previsto por decreto, trazendo consigo grandes limitações para a realização das atividades previstas. Foi realizada uma revisão bibliográfica sobre o funcionamento dos tipos de sensores utilizados no projeto, assim como sobre diabetes, hipoglicemia e efeitos da hipoglicemia relacionados ao batimento cardíaco, temperatura e umidade do corpo. Criou-se o software e uma modelagem do dispositivo para prever episódios hipoglicêmicos, porém não foi possível sua prototipação física devido à falta de recursos e a impossibilidade de testes funcionais com pacientes.

**Palavras-Chave: Diabetes. Hipoglicemia. Microcontrolador PIC.**

## SUMÁRIO

<b>INTRODUÇÃO</b>	<b>5</b>
<b>OBJETIVOS</b>	<b>6</b>
<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>7</b>
<b>MÉTODO</b>	<b>14</b>
<b>RESULTADOS E DISCUSSÃO</b>	<b>166</b>
<b>CONSIDERAÇÕES FINAIS</b>	<b>1919</b>
<b>REFERÊNCIAS</b>	<b>21</b>
<b>ANEXO A – Código de captação de dados e envio para Excel</b>	<b>23</b>
<b>ANEXO B – Código de ativação do sistema de controle</b>	<b>28</b>

## INTRODUÇÃO

O uso de tecnologia na prevenção, diagnóstico e tratamento de doenças, por meio de sensores, atuadores e equipamentos eletrônicos em geral, permite à medicina uma melhor atuação na vida dos pacientes. Tratamentos para disfunções como o Diabetes Mellitus, que traz consigo um descontrole glicêmico, trazem a cada dia inovações e melhorias, com o uso, por exemplo, do sensor de glicemia. A hipoglicemia é um dos sintomas que mais afeta diabéticos no mundo inteiro, com seus mais diversos sintomas, havendo grande necessidade de prevenção e controle dela. A possibilidade de se fazer um prognóstico do quadro hipoglicêmico mais acurado e prévio ao obtido por sensores glicêmicos, por meio da relação entre a variação da frequência cardíaca, diminuição da temperatura corpórea, aumento da umidade da pele e a queda na taxa de açúcar no sangue, é uma oportunidade de melhoria na qualidade de vida dos portadores de Diabetes Mellitus tipo 1, condição crônica na qual o corpo produz pouca ou nenhuma insulina. O dispositivo proposto neste projeto fará a leitura da variação de frequência cardíaca, temperatura e umidade corpórea, processando-os graficamente e revelando sua relação com o quadro hipoglicêmico em diabéticos.

De acordo com as Diretrizes da Sociedade Brasileira de Diabetes, "o diabetes mellitus (DM) consiste em um distúrbio metabólico caracterizado por hiperglicemia persistente, decorrente de deficiência na produção de insulina ou na sua ação, ou em ambos os mecanismos, ocasionando complicações em longo prazo" [1]. Dentre os tipos de Diabetes, tem-se Diabetes Tipo 1, Diabetes Tipo 2, Pré Diabetes e Diabetes Gestacional, sendo o foco desse projeto o Diabetes Tipo 1. Segundo dados da IDF (International Diabetes Federation), a doença atingia 16,8 milhões de brasileiros em 2019, com o Brasil atingindo a 5ª posição no Rank mundial de Diabetes, atingia 463 milhões de pessoas no mundo com tendência de que esse número aumente a cada ano, chegando a uma previsão de 700 milhões de pessoas com diabetes até 2045 [2].

Um dos métodos desenvolvidos para prevenção de hipoglicemias, é o uso dos chamados Sensores de CGM – “Continuous Glucose Monitor” – que, conectados ao corpo por uma pequena cânula de silicone e em contato com o sangue, medem a glicemia e enviam os dados medidos para uma bomba de insulina – equipamento responsável por injetar insulina no organismo continuamente e realizar correções quando necessário, sendo substituto da caneta de insulina – ou monitor continuamente, reduzindo a quantidade de medições de

ponta de dedo para análise da quantidade de glicose no sangue. O CGM, trabalhando em conjunto com a bomba de insulina, de fato reduz a quantidade de hipoglicemias em diabéticos, mas é ainda um equipamento caro e, no Brasil, de difícil acesso. Busca-se, dessa forma, e com o projeto de pesquisa aqui proposto, uma alternativa ao uso do CGM, mais barata e não invasiva e tão eficiente quanto a existente.

Devido ao fato de a hipoglicemia levar a manifestações de liberação do Sistema Simpático, que por sua vez, afeta a variabilidade de frequência cardíaca, liberação de suor e alterações nos níveis de temperatura corporal, existe a possibilidade de se estabelecer uma relação entre esses sintomas e a hipoglicemia. A ideia é aproveitar essas interações, buscando as referências que relacionam a hipoglicemia com a variabilidade de frequência cardíaca, umidade e temperatura corpórea, fazendo a aquisição e processamento desses sinais biológicos, estabelecendo-se uma relação entre eles. Para tanto, será desenvolvido, como interface, um sistema eletrônico controlado por PIC. Ele será o microcontrolador usado nesse projeto para elaboração da leitura de dados coletados pelos sensores, Pulse Sensor e SHT15, e elaboração dos gráficos de retorno por meio do método de Poincaré, que permite a clara visualização da variação de tempo entre cada batimento cardíaco e a variação de umidade e temperatura corpórea.

## **OBJETIVOS**

O **objetivo geral** deste projeto é desenvolver e avaliar o desempenho de um dispositivo eletrônico capaz de fazer o prognóstico de hipoglicemias noturnas em diabéticos tipo 1, por meio da variabilidade da frequência cardíaca junto com umidade e temperatura da pele. O projeto visa analisar a relação entre hipoglicemias e a taxa de frequência cardíaca na região do dedo médio juntamente com alterações dos níveis de umidade e temperatura corpórea, assim como alcançar um método mais eficiente para a previsão de episódios de risco na vida do paciente diabético, trazendo um dispositivo que visa um conforto maior para o paciente, já que o mesmo não é invasivo, diferente dos métodos já existentes, em especial, o CGM.

### **Os objetivos específicos:**

- Realizar uma revisão bibliográfica sobre o funcionamento dos tipos de sensores utilizados no projeto, assim como sobre diabetes, hipoglicemia, efeitos da hipoglicemia relacionados a sinais biológicos citados anteriormente e elaboração do sistema com microcontrolador.
- Elaborar software de controle dos sensores por meio de ferramentas usuais e compatíveis com PIC, como por exemplo Protheus e MPLAB de modo a se obter uma boa medição dos sinais captados pelos sensores.
- Elaborar um programa capaz de converter e processar os sinais cardíacos de HRV no formato digital em Poincaré Plots.
- Elaborar um hardware centrado na arquitetura PIC e seus periféricos;
- Analisar e testar a funcionalidade do programa e circuito desenvolvido, de modo a verificar a eficiência e realizar possíveis ajustes conforme necessidade.
- Projetar um dispositivo confortável para o paciente.
- Realizar testes para verificação em Diabéticos tipo 1 voluntários para levantamento de dados probabilísticos e testes de desempenho.
- Levantar dados que indiquem relação entre hipoglicemias, variabilidade de frequência cardíaca e alteração da umidade e temperatura da pele.
- Comparar resultados obtidos com os obtidos pelo CGM para analisar vantagens, desvantagens e levantar possíveis melhorias a serem feitas.

## **FUNDAMENTAÇÃO TEÓRICA**

Aqui são apresentados fundamentos necessários para o embasamento teórico deste trabalho. Os conceitos sobre o Diabetes Mellitus e seus tipos, hipoglicemia, microcontrolador PIC e sensores utilizados neste projeto estão representados abaixo de forma a tentar facilitar o entendimento sobre eles.

### **1. Diabetes**

A insulina é um hormônio naturalmente produzido pelo corpo humano, responsável por controlar a quantidade de glicose – açúcar – no sangue e é essencial para que haja

aproveitamento dela para produção de energia. Com sua ausência, o processo de quebra da glicose não é cumprido, gerando assim um acúmulo de glicose no sangue e ao invés de ser armazenada como gordura, terá seu excesso eliminado na urina [3]. O hormônio da Insulina serve como uma “chave” que ao se ligar ao receptor de insulina abre o canal para que a glicose entre no transportador de glicose e assim gere energia na célula. Esse processo pode ser observado na figura abaixo.

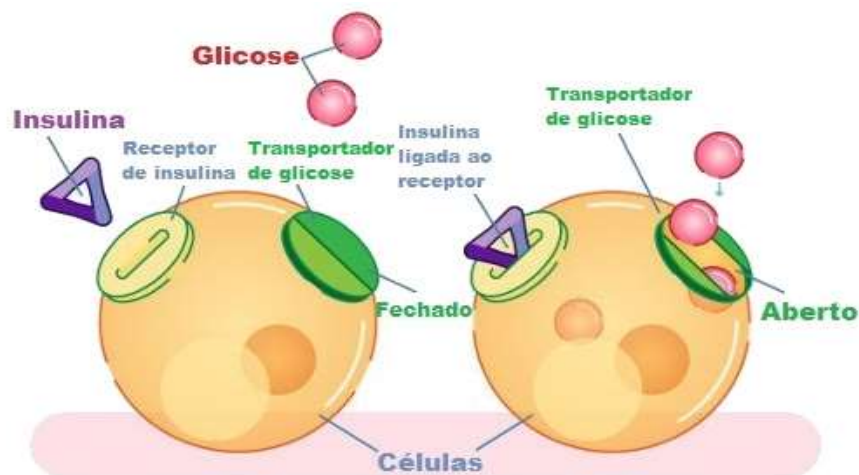


Figura 1: Funcionamento do hormônio Insulina.

(Fonte: “<https://blog.biologicus.com.br/insulina-voce-sabe-para-que-serve/>”)

De acordo com as Diretrizes da Sociedade Brasileira de Diabetes, "o diabetes mellitus (DM) consiste em um distúrbio metabólico caracterizado por hiperglicemia persistente, decorrente de deficiência na produção de insulina ou na sua ação, ou em ambos os mecanismos, ocasionando complicações em longo prazo" [1]. Dentre os tipos de Diabetes, tem-se Diabetes Tipo 1, Diabetes Tipo 2, Pré Diabetes e Diabetes Gestacional, sendo o foco deste projeto o Diabetes Tipo 1 (DM1).

O DM1 também conhecido como Diabetes infanto-juvenil, tem esse nome por se apresentar mais frequentemente em crianças e adolescentes. O corpo da pessoa com DM1 não produz insulina, ou a produz em quantidade insuficiente, devido a uma reação auto imune de destruição das células beta nas ilhotas do pâncreas [2], responsáveis pela produção dessa insulina, não havendo, então, aproveitamento da glicose, ou seja, o corpo não utiliza o açúcar ingerido na alimentação, fazendo com que o nível de glicose no sangue aumente. A este fator dá-se o nome de hiperglicemia – excesso de glicose no sangue – podendo ser



causado por fatores como, por exemplo, a falta de exercícios físicos, alimentação inapropriada e a dosagem incorreta da insulina [4]. É importante apontar que este último fator citado é algo frequente na vida de um diabético tipo 1, devido a necessidade de realizar o controle dos níveis glicêmicos a partir do uso de insulina artificialmente produzida, que em dosagens incorretas podem causar tanto uma hiperglicemia quanto uma hipoglicemia.

## **2. Hipoglicemia e CGM**

A chamada hipoglicemia, ou falta de açúcar no sangue, ocorre por motivos diversos, dentre eles a falta de alimentação devida, o excesso de insulina aplicada, ou mesmo a prática excessiva de exercícios físicos. De acordo com o Ministério da Saúde a hipoglicemia é caracterizada quando os níveis glicêmicos caem para valores abaixo de 60 a 70 mg/dL, levando a sintomas neuroglicopênicos (fome, tontura, fraqueza, dor de cabeça, confusão, coma, convulsão) e a manifestações de liberação do sistema simpático (sudorese, taquicardia, apreensão, tremor) [5]. Em contraste a hipoglicemia, se tem a hiperglicemia que para a SBD (Sociedade Brasileira De Diabetes) é caracterizada por índices glicêmicos acima de 240mg/dL de sangue e afeta seriamente o corpo a longo prazo (doenças cardíacas, insuficiência renal, problemas oculares, depressão, danos cerebrais, etc), os sintomas da hipoglicemia afetam o corpo no momento do episódio hipoglicêmico, sendo vista, por tal, como uma situação de urgência a ser tratada com a ingestão de açúcar, sendo recomendado o consumo de carboidratos simples em uma quantidade de 15 a 20 gramas, podendo ser uma colher de sopa de açúcar dissolvida em um copo com água, 200ml de refrigerante comum (não diet), um copo de suco de laranja integral ou até uma colher de sopa de mel, sendo o mel não permitido para crianças menores de 1 ano [4, 6].

Um dos métodos desenvolvidos para prevenção de hipoglicemias, é o uso dos chamados Sensores de CGM – “Continuous Glucose Monitor” – que, conectados ao corpo por uma pequena cânula de silicone que em contato com o sangue, medem a glicemia e enviam os dados medidos para uma bomba de insulina – equipamento responsável por injetar insulina no organismo continuamente e realizar correções quando necessário, sendo substituto da caneta de insulina – ou monitor continuamente, reduzindo a quantidade de medições de ponta de dedo para análise da quantidade de glicose no sangue. Vale citar que o próprio uso da bomba de insulina já reduz o problema dos picos de insulina notados no uso de canetas de insulina, conseqüentemente melhorando o quadro de hipoglicemias causadas por estes picos.

A figura 2 abaixo mostra o CGM, acoplado ao corpo, vendido pela Medtronic, conectado com a bomba de insulina funcionando em conjunto como exposto anteriormente.

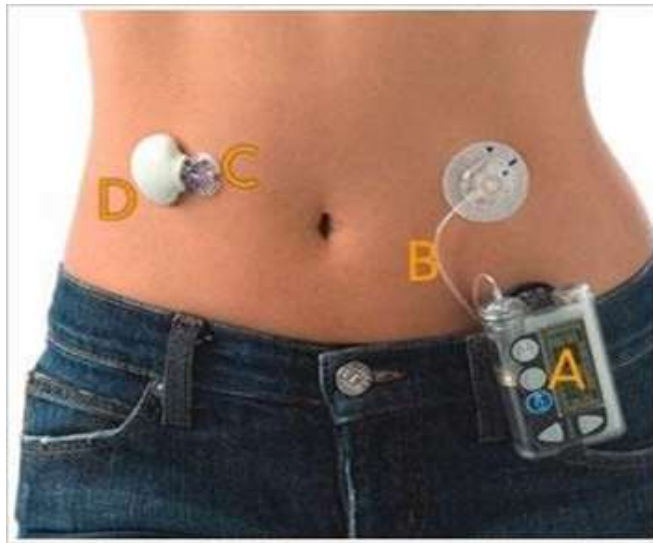


Figura 2: CGM e Bomba de Insulina Medtronic (Fonte: “<http://www.medtronicdiabetes.com.br/acerca-do-produto/paradigm-veo/monitorizacao-continua-da-glicose.htm>”l)

A Figura 3 mostra o gráfico formado pela bomba ao receber os dados do CGM (sendo estes enviados, através de Rádio Frequência a cada 20 minutos), assim como o último valor no histórico.



Figura 3: Bomba de Insulina com gráfico de variação glicêmica (Fonte: “<https://www.biabetica.com/single-post/2017/02/20/O-que-e-cgm?lightbox=dataptem-j8evd4ra>”)

### 3. Sinais Biológicos e a Hipoglicemia

No estudo realizado por Simon Cichosz et al. [7], é analisada a possibilidade de prever hipoglicemias em diabéticos por meio da relação vista entre a hipoglicemia e a variabilidade de frequência cardíaca. Esse método utiliza um sensor de HRV (Heart Rate Variability) para analisar a variabilidade de frequência cardíaca em momentos de estabilidade glicêmica no paciente e em momento de hipoglicemia. Variabilidade de frequência cardíaca faz referência a variação natural no intervalo de tempo entre cada batimento cardíaco (Inter-beat Interval, ou IBI), e é um valor que flutua no tempo, não sendo constante como as batidas por minuto (BPM), que são médias de algumas batidas no tempo, por isso mais regulares. O IBI é afetado pelos Sistemas Nervosos Simpático e Parassimpático, por condições médicas, psicológicas ou mesmo pela respiração. Dessa forma, a hipoglicemia, que leva a manifestações de liberação desse mesmo Sistema Simpático (sudorese, taquicardia, apreensão, tremor), afeta também esse intervalo, de modo que existe a possibilidade de se estabelecer uma relação entre os dois. O Inter Beat Interval (IBI), também conhecido como intervalo RR, é calculado como o período entre dois batimentos cardíacos consecutivos, considerando-se o momento de pico do batimento cardíaco (ponto R na curva QRS). Para um ser humano em estado normal, espera-se que o intervalo esteja na faixa de 650 ms a 1200 ms entre batimentos cardíacos, equivalente aos valores de 50 a 100 batimentos por minuto [8].

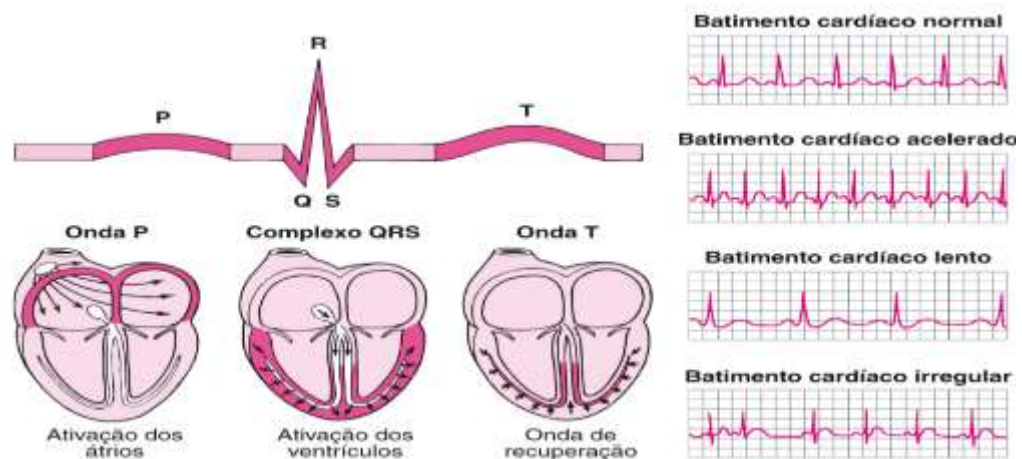


Figura 4: Eletrocardiograma (ECG)

(Fonte: Michael J. Shea)

No estudo realizado por V. Coulic et al. [9], utilizando sensores de temperatura introduzidos no reto do paciente que, dependendo da variação da temperatura corporal

medida, enviam sinais para que haja a liberação de um volume de insulina ajustável, por meio da bomba de insulina acoplada ao seu corpo, com o intuito de melhorar o controle dos níveis de glicose em pacientes diabéticos. Com base nesse estudo e considerando a premissa de que a temperatura corporal está relacionada com a variação dos níveis de glicose no sangue, foi decidido que a temperatura, juntamente com a umidade da pele (suor) – que, como exposto anteriormente, também é uma manifestação de liberação do Sistema Simpático – serão sinais biológicos utilizados neste projeto com o intuito de tornar a previsão de episódios hipoglicêmicos mais precisa.

Assim, os três sinais biológicos de controle que possuem relação com a hipoglicemia e que serão utilizados como base para esse projeto, são a Variabilidade de Frequência Cardíaca, a Temperatura e Umidade, que serão medidas através dos sensores Pulse Sensor e SHT15 de maneira menos invasiva que nos estudos aqui citados.

#### **4. Microcontrolador PIC**

O Microcontrolador PIC se trata de um circuito integrado, onde em um único dispositivo possui um sistema digital programável. Ele possui uma CPU (Central Processor Unit ou Unidade Central de Processamento) que fará a interpretação das instruções programadas; uma memória PROM (Memória Programável Somente para Leitura) que irá memorizar de maneira permanente as instruções do programa; Memória RAM (Memória de Acesso Aleatório), memoriza as variáveis utilizadas pelo programa; Possui também uma série de Linhas I/O (entrada e saída), que fará o controle dos dispositivos externos ou receber pulsos de sensores. Também conta com dispositivos auxiliares, geradores de clock, bus, contador [10]. Com todos esses recursos, se torna mais vantajoso sua utilização.

O PIC18F4550, será utilizado neste trabalho, com 40 pino cujas funcionalidades podem ser estabelecidas por programação, conexão USB e mais alguns itens úteis, como um botão de reset. O PIC18F4550 contém diversos recursos dentre eles, possui uma CPU de 8 bits, tensão operacional de +4,0 a +5,5V ( tendo como 5,5V, sua tensão máxima absoluta), tem um total de 35 pinos de entrada e saída programáveis, interfaces de comunicação, módulo ACD (Automação em Coleta de Dados) onde se tem 13 canais, um módulo temporizador, comparadores analógicos, canais PWM (Pulse Width Modulation), oscilador interno e temporizador de Watchdog com oscilador on-chip separado. Conectado ao computador, permite ao usuário que seja feita sua programação para as mais diversas funcionalidades [11].

## **5. Sensores utilizados (Pulse Sensor e SHT15)**

Um sensor diz respeito a um dispositivo sensível a alguma forma de energia que se encontra no ambiente, relacionando informações sobre uma grandeza que precisa ser medida.

Os sensores podem ser divididos em dois grupos, um de sensores analógicos, que poderá assumir qualquer valor no seu sinal de saída ao longo do tempo, desde que esteja dentro de seu limite de operação. E os sensores digitais, que poderão assumir somente dois valores em seus sinais de saída ao longo do tempo, que podem ser interpretados como um ou zero. Em questões de sensibilidade, temos que o sensor analógico, está diretamente ligado à relação entre uma variação na grandeza que está sendo medida e a variação medida fornecida pelo instrumento. Um sensor sensível é aquele que fornece uma variação na saída para uma pequena variação da grandeza medida. Todos os sensores possuem um erro de leitura que poderá ser diferente, conforme a grandeza que será medida, as condições de operação do sensor, estado de conservação e sua qualidade. Uma das grandezas que se é encontrado certas dificuldades para que seja medida com precisão, é a umidade, a qual fará parte deste projeto. Levando isso em consideração, a maior parte dos sensores comerciais apresentam um erro de leitura entre 2% e 10% [12].

O Pulse Sensor, que trata do sensor que será utilizado para medir os batimentos cardíacos, é um sensor plug-and-play desenvolvido pela empresa World Famous Electronics llc. desde 2011, que permite uma conexão com o microcontrolador PIC. Possui diversas aplicações, dentre elas, a medição de sinais de HRV. Sua praticidade e baixo custo, além da conexão com PIC fazem dele um método de sensoriamento bom e eficiente [13].

O sensor utilizado para que sejam medidas as grandezas de temperatura e umidade, será o SHT15, um sensor de alta precisão e excelente estabilidade que integra em uma placa dois sensores em um único chip, além do processamento de sinal. Para que seja medida a umidade relativa é usado um do tipo capacitivo exclusivo, enquanto a temperatura é medida por um sensor de intervalo de banda. Os sensores são acoplados, como já citado antes, em um conversor analógico / digital de 14 bits. Tendo isso em vista, é obtido uma qualidade de sinal superior [14,15].

## MÉTODOLOGIA

**Tipificação:** O trabalho aqui apresentado se caracteriza como uma pesquisa exploratória de natureza aplicada, onde o problema será abordado de forma qualitativa e os procedimentos técnicos serão experimentais.

**Local de pesquisa:** Devido ao trabalho estar voltado tanto para a área da engenharia da computação, quanto para a área de saúde, o local principal da pesquisa estava previsto para ser o Labocien do Centro Universitário de Brasília – UniCEUB, que com a disponibilidade de equipamentos pertencentes ao laboratório, possibilitaria o desenvolvimento do dispositivo visado na realização deste projeto. Entretanto, com a impossibilidade de acesso ao Labocien perante a pandemia do Covid-19, o local de pesquisa foi a residência de ambos os pesquisadores, sem o acesso aos equipamentos necessários.

**Objeto de estudo:** O objeto de estudo deste projeto é a hipoglicemia em portadores de Diabetes Tipo 1.

**Delimitação e universo da amostra:** Inicialmente buscava-se para a realização deste trabalho o número de 10 pessoas (podendo-se aumentar este número futuramente). O critério de inclusão dos voluntários na pesquisa era a existência de diagnóstico médico que comprove que a pessoa é portadora de Diabetes Tipo 1, além disso deverá ter um aparelho CGM para coletar dados de comparação. Porém, devido a pandemia do Covid-19 os testes não puderam ser realizados, principalmente levando em consideração que portadores de Diabetes estão inclusos no grupo de risco.

**Instrumento de coleta ou de geração de dados:** Os dados para a pesquisa seriam coletados por meio do dispositivo que será desenvolvido ao longo do projeto juntamente com um aparelho CGM já existente, permitindo assim uma comparação dos dados coletados por cada um dos dispositivos. Porém, devido a impossibilidade de realizar a coleta dos dados, essa comparação não foi possível.

### **Procedimentos metodológicos:**

A **primeira etapa** do estudo (pré-pesquisa) é constituída pelo estudo de artigos, teses e dissertações disponíveis em bases de acesso público e seleção, dentre eles, dos mais relevantes publicados entre janeiro de 1990 e 2019. Os arquivos de maior relevância para este projeto foram encontrados por meio dos sites de pesquisa Google Academic e IEEE Xplore.

A **segunda etapa** foca no estudo aprofundado da bibliografia encontrada. Através da leitura dos documentos encontrados, foi realizada a fundamentação teórica para cada parte do dispositivo proposto, assim como do software e da comparação entre ele e os dispositivos vendidos no mercado atualmente. Seguir-se-á o princípio de acesso aberto ao conhecimento, assim como às tecnologias (software e hardware) utilizados para desenvolvimento do produto.

A **terceira etapa** consiste no desenvolvimento do Módulo 1, sendo este o sistema de captação dos sinais biológicos por meio dos sensores, de forma que haja boa conexão destes com o corpo do paciente.

Na **quarta etapa** será feito o desenvolvimento do Módulo 2, que se baseia no desenvolvimento da placa de conexão entre o PIC e os sensores, de modo a deixar o equipamento o mais confortável e portátil possível.

Na **quinta etapa** será feita a adaptação do software para o desenvolvimento do Módulo 3 que, por sua vez, será utilizado para funcionamento dos sensores em comunhão com o microcontrolador, de modo a se obter os sinais em formato gráfico para comparação.

A **sexta etapa** se baseia na preparação e início dos testes comparativos, com a análise de um especialista na área, para se avaliar o bom funcionamento dos sinais recebidos, e realizar ajustes para que se obtenha resultados mais precisos. Contudo, devido a impossibilidade da realização dos testes por conta do Covid-19, esta etapa não pôde ser realizada.

Por fim, na **sétima etapa**, seriam feitos testes em voluntários portadores de Diabetes Tipo 1, para análise do quadro de hipoglicemia e estabelecimento da relação entre este e os sinais obtidos. Nessa etapa, seriam obtidos os dados necessários para levantamento do quadro de eficiência e confiabilidade do dispositivo elaborado, assim como do obtido pelo uso de sensores CGM, e a comparação de ambos. Com a obtenção desta comparação seria possível propor possíveis melhorias e futuras aplicações para o dispositivo proposto, encerrando, assim, o trabalho. Esta etapa também foi impossibilitada devido a pandemia do Covid-19.

Para a realização dos testes seria feito um monitoramento dos sinais biológicos dos pacientes, para a averiguação da precisão do dispositivo desenvolvido, prezando sempre pelo conforto do paciente, em horários de disponibilidade dos mesmos e com autorização dos seus médicos responsáveis. Está se tratando de uma parte fundamental do projeto, tendo em vista

que a proposta é o desenvolvimento de um dispositivo que irá trazer consigo um conforto para os portadores de Diabetes do Tipo 1 e sem os testes, não foi possível a afirmação de um dispositivo eficaz e confiável.

Em complemento à metodologia apresentada seriam utilizados equipamentos para testes e desenvolvimento do protótipo tais como:

- Osciloscópio digital;
- Analisador de sinais;
- Fonte de alimentação;
- Softwares para simulação de circuitos elétricos Proteus, Fritzing.
- Multímetro digital;
- Kit PIC para montagem do protótipo;

## **RESULTADOS E DISCUSSÃO**

Com a pesquisa realizada e tomando como base trabalhos já existentes na área, observa-se que existe a possibilidade de se estabelecer uma relação entre episódios hipoglicêmicos e sinais biológicos afetados pela manifestação do Sistema Simpático (sudorese, taquicardia, apreensão, tremor). Partindo deste princípio, buscou-se o desenvolvimento de um sistema que integrasse a captação dos sinais biológicos descritos na seção de Fundamentação Teórica, sendo eles IBI, temperatura corpórea e umidade da pele (sudorese).

A codificação foi toda feita com linguagem de programação para Arduino e com o uso do Pinguino, uma ferramenta que permite programar PIC com a mesma abordagem com que o Arduino é programado. Foi optado pelo uso do Pinguino por ser compatível com o PIC18F4550 e sua praticidade, visto que é composto por uma IDE onde estão todos os artefatos necessários ao desenvolvimento, como editor de código, comandos para compilação e gravação de código no PIC, além de depuração via USB [16]. Para o desenvolvimento do software foram tomados como base os seguintes parâmetros:

- IBI com intervalo entre 650 ms e 1200 ms para quando o paciente está em estado normal;
- Umidade (Intervalo a ser estudado);
- Temperatura (Intervalo a ser estudado).



Devido aos parâmetros de umidade e temperatura não terem um intervalo pré definido por meio de estudos já realizados, devem ser coletados dados a partir de um universo amostral, e a partir destes dados poderá ser determinado um intervalo aceitável à ser utilizado no software como parâmetro para melhor previsão de um episódio hipoglicêmico.

Devido a necessidade da coleta desses dados, foram desenvolvidos dois códigos, sendo um para a captação dos sinais biológicos e envio dos mesmos para o Excel (Anexo A deste documento) e outro responsável pelo acionamento do sistema de alerta, inicialmente usando somente o IBI e posteriormente incluindo a umidade e temperatura que teriam seus intervalos definidos após estudo (Anexo B deste documento).

Para o desenvolvimento do circuito elétrico, feito no Proteus, que deve ser utilizado como base para elaboração do protótipo físico, foram utilizados os seguintes sensores e componentes utilizados para captação dos sinais biológicos:

- Pulse Sensor;
- SHT15
- PIC18F4550
- Resistores
- Potenciômetro
- LED
- Buzzer

O Pulse Sensor, responsável pela captação da frequência cardíaca, possui 3 pinos sendo eles GND, VCC e Data. O pino GND foi ligado ao terra, o VCC foi conectado a uma entrada de 5v, pois segundo recomendações do fabricante tem funcionamento com uma tensão de 3v a 5v. O pino Data, responsável pelo envio das informações para o microcontrolador, foi conectado a porta 35 do PIC, que também pode ser descrita como porta RB2, podendo ser configurada como porta de entrada analógica.

O sensor de temperatura e umidade, que possui dois pinos de comunicação, sendo eles SCK e DATA. O pino SCK responsável pelo controle de clock do sensor, foi conectado à porta 34 também descrita como RB1/SCK. Já o DATA foi conectado à porta 33 também descrita como RB0, seguindo recomendações do fabricante foi colocado um resistor de 10kΩ. Para a confecção do protótipo físico, também por recomendações do fabricante como mostrado na Figura 5, deverá ser adicionado um capacitor de 10nF, tendo este capacitor ligado o mais próximo possível do sensor, que terá por finalidade atenuar interferências oriundas da fonte

de alimentação. O resistor de 10kΩ funcionará como um pull-up no barramento de dados. Também é possível a utilização de um resistor no barramento de sincronização com a mesma finalidade.

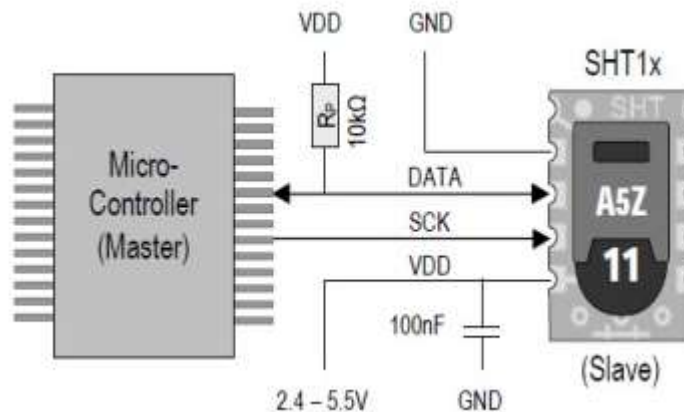


Figura 5: Recomendação de ligação para sensores SHT1x  
(Fonte: Folha de dados do fabricante)

O microcontrolador PIC18F4550, tem por finalidade fazer o controle do sistema por meio do Software desenvolvido, que ficará acoplado em sua memória.

O resistor conectado em série com o LED, tem por finalidade controlar a tensão que será distribuída, evitando assim que ele venha a queimar.

O potenciômetro que foi ligado a porta RA0 tem a função regular a intensidade do buzzer, que por sua vez foi conectado à porta RA1, ambas as portas podem ser configuradas como entrada ou saída, e foram conectados ao terra.

A Figura 6 que se encontra a seguir mostra a modelagem do circuito, feita utilizando a ferramenta Proteus.

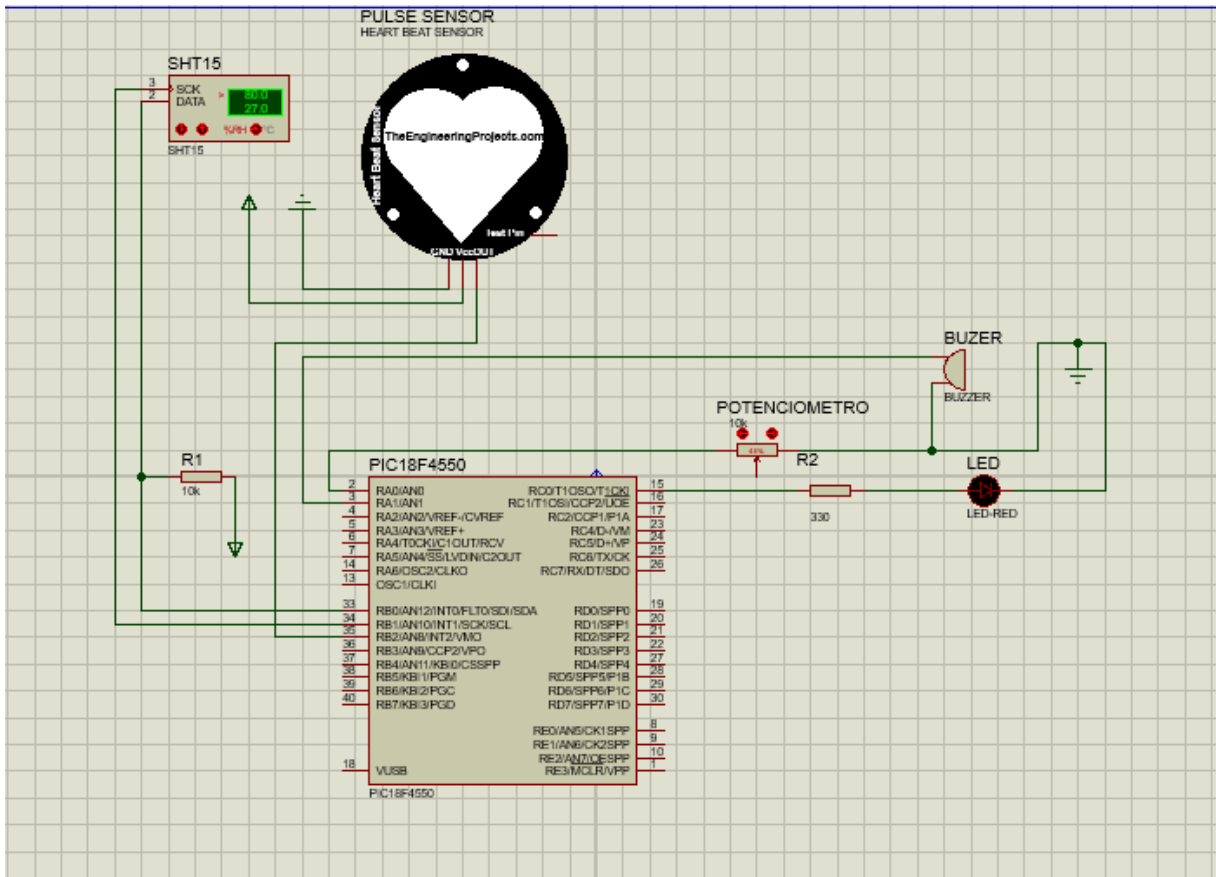


Figura 6: Modelagem do circuito

(Fonte: Autor)

## CONSIDERAÇÕES FINAIS

A hipoglicemia é um dos fatores de grande risco a vida dos diabéticos, portanto é necessário o uso de aparelhos de grande eficácia na predição e tratamento dessa condição. Esse ponto é muito importância nos estudos da área da tecnologia juntamente à área da saúde. Buscou-se com esse estudo desenvolver e avaliar o desempenho de um dispositivo eletrônico capaz de fazer o prognóstico de hipoglicemias noturnas em diabéticos tipo 1, por meio da variabilidade da frequência cardíaca junto com umidade e temperatura da pele.

No decorrer do período de realização da pesquisa surgiram empecilhos devido ao Covid-19, como a falta de acesso aos materiais e equipamentos que seriam utilizados no Labocien do UniCEUB e a impossibilidade de realização dos testes, que acarretaram o não cumprimento dos objetivos propostos.

O UNICEUB no dia 16 de março de 2020 publicou um comunicado de suspensão das atividades educacionais em respeito ao decreto 40.520, e a partir deste momento iniciaram-se as atividades remotas, porém permanecendo a impossibilidade de acesso ao campus por tempo indeterminado. No dia 16 de julho de 2020 o UniCEUB mandou um comunicado oficial reforçando a realização das atividades do segundo semestre sendo mediadas por recursos tecnológicos, de acordo com a Portaria nº. 544 publicada pelo MEC no dia 16 de junho de 2020.

Parte das atividades previstas inicialmente foram cumpridas com êxito. Entre elas se encontram o levantamento teórico para sustentação do trabalho, a modelagem do circuito que seria utilizado como base para a implementação física do protótipo e a codificação básica a ser utilizada para a coleta dos dados e acionamento do sistema.

Houve um atraso na aquisição dos materiais o que impossibilitou o início da prototipação física no período previsto, e logo em seguida iniciou-se o período de pandemia e por conta disso as atividades que necessitavam do uso do espaço do Labocien não puderam ser realizadas. Quaisquer testes previstos com voluntários não foram realizados, visto que o grupo a participar (portadores de Diabetes) foi classificado como parte do grupo de risco e prezando pela saúde e segurança de todos envolvidos não seria condizente a realização dos testes [17].

Considerando tudo o que foi exposto até agora, é cabível algumas considerações sobre próximos trabalhos a serem desenvolvidos com base na pesquisa já realizada. O ponto principal a ser trabalhado futuramente é a implementação física do aparelho proposto, usando como base a modelagem do circuito e a codificação criada durante este projeto. Após se ter um aparelho físico, deve-se realizar testes para ajustar qualquer incoerência entre o software e o hardware para que então tenham início os testes em voluntários para a coleta de dados. Com os dados coletados é preciso realizar uma análise deles para ver a possibilidade de um intervalo de temperatura e umidade a ser usado para o acionamento do sistema de alerta. Por fim fazer ajustes de conforto e confiabilidade do sistema. Ao final destes passos é necessário realizar um estudo comparativo sobre os dados coletados através do protótipo e os dados coletados por meio do CGM, para averiguar a efetividade do sistema criado.

## REFERÊNCIAS

- [1] Sociedade Brasileira de Diabetes (2017) **Diretrizes da Sociedade Brasileira de Diabetes 2017-2018.**  
Link<<https://www.diabetes.org.br/profissionais/images/2017/diretrizes/diretrizes-sbd-2017-2018.pdf>>
- [2] International Diabetes Federation (2019) **IDF Diabetes Atlas - 9th Edition.**  
Link<<https://diabetesatlas.org/en/resources/>>
- [3] OLIVEIRA, Rebeca (2018) **SISTEMA PARA PROGNÓSTICO E ANÁLISE DE EPISÓDIOS HIPOGLICÊMICOS EM DIABÉTICOS TIPO 1 BASEADO EM SENSOR DE VARIABILIDADE DA FREQUÊNCIA CARDÍACA CONTROLADO POR ARDUINO.**  
Link<<https://www.publicacoesacademicas.uniceub.br/pic/article/view/5779>>
- [4] Sociedade Brasileira de Diabetes (2019) **Hiperglicemia**  
Link<<https://www.diabetes.org.br/publico/diabetes/hiperglicemia>>
- [5] Departamento de Atenção Básica, Secretaria de Atenção à Saúde, Ministério da Saúde (2006) **Cadernos de Atenção Básica - n.º 16.**  
Link<[http://bvsmms.saude.gov.br/bvs/publicacoes/diabetes\\_mellitus\\_cab16.pdf](http://bvsmms.saude.gov.br/bvs/publicacoes/diabetes_mellitus_cab16.pdf)>
- [6] Sociedade Brasileira de Diabetes (2019) **Hipoglicemia**  
Link<<https://www.diabetes.org.br/publico/diabetes/hipoglicemia>>
- [7] Cichosz, S., Frystyk, J., Hejlesen, O.K., Tarnow, L., Fleischer, J. (2014) **A Novel Algorithm for Prediction and Detection of Hypoglycemia Based on Continuous Glucose Monitoring and Heart Rate Variability in Patients With Type 1 Diabetes.** (n.d.). Journal of Diabetes Science and Technology, 1. Link<<https://doi.org/10.1177/1932296814528838>>
- [8] LAPA, Eduardo. (2016) **A frequência cardíaca normal fica entre 60 e 100 bpm, certo? Errado!** Link<<https://cardiopapers.com.br/a-frequencia-cardiaca-normal-fica-entre-60-e-100-bpm-certo-errado/>>
- [9] V.Coulic, V.Novikov, J.Devriendt, C.Gillet, A.Bodson, E. DeKoster (2007) **Use of Temperature Gradient Measuring Device in Monitoring of Diabetic and Critically Ill Patients.** Link<<https://ieeexplore.ieee.org/document/4353087>>
- [10] Trevesian, Pedro **MICROCONTROLADORES**  
Link<[https://www.radioamadores.net/files/microcontroladores\\_pic.pdf](https://www.radioamadores.net/files/microcontroladores_pic.pdf)>

[11] **Especificações do microcontrolador PIC18F4550**

Link<<https://components101.com/microcontrollers/pic18f4550-pin-diagram-features-datasheet>>

[12] THOMAZINI, Daniel. ALBUQUERQUE, Pedro (2011) **Sensores Industriais**

**Fundamentos e Aplicações.**

[13] **Especificações do PulseSensor.**

Link<[www.pulsesensor.com](http://www.pulsesensor.com)>

[14] **Especificações do SHT15.** Link<<https://pt.farnell.com/sensirion/sht15/sensor-humidity-temp-3-3v/dp/1590512>>

[15] **Especificações do SHT15.** Link<<https://multilogica-shop.com/placa-com-sensor-de-umidade-e-temperatura-sht15>>

[16] CURVELLO, André (2015) **Pinguino: Programando PIC como Arduino.**

Link<<https://www.embarcados.com.br/pinguino-programando-pic-como-arduino/>>

[17] Medeiros e Wada (2020) **Guia de Vigilância Epidemiológica - Emergência de Saúde Pública de Importância Nacional Pela Doença Coronavírus 2019.**

Link<[https://portalarquivos.saude.gov.br/images/af\\_gvs\\_coronavirus\\_6ago20\\_ajustes-finais-2.pdf](https://portalarquivos.saude.gov.br/images/af_gvs_coronavirus_6ago20_ajustes-finais-2.pdf)>

## ANEXO A – Código de captação de dados e envio para Excel

```
const int LED = 15;
const int BUZZER = 3;
const int POT = 2;
const int pulsePin = 35;
const int clockPin = 34; // pin para clock
const int dataPin = 33; // pin para dado
int ROW = 0;
int LABEL = 1;
int val = 0;
int comandoTemperatura = B00000011; // comando usado para ler temperatura
int comandoUmidade = B00000101; // comando usado para ler umidade
int ack; // variavel para erros
int valT;
int valU;
float temperatura;
float humidade;
volatile int BPM; //Declaradas como volatile por serem usadas
no ISR e em outras partes do codigo
volatile int Signal;
volatile int IBI = 600;
volatile boolean Pulse = false;
volatile boolean QS = false;
volatile int rate[10];
volatile unsigned long sampleCounter = 0;
volatile unsigned long lastBeatTime = 0;
volatile int P = 80;
//valor de pico
volatile int T = 80;
//valor through
volatile int thresh = 80; //tresh
inicializado em 512 ms (metade da amplitude analogica)
volatile int amp = 100;
volatile boolean firstBeat = true;
volatile boolean secondBeat = false;

void setup(){
  pinMode(LED, OUTPUT);
  pinMode(Buzzer, OUTPUT);
  pinMode(POT, INPUT); // Por padrão porta analógica já é
  INPUT
  pinMode(pulsePin, INPUT);
  pinMode(clockPin, INPUT);
  pinMode(dataPin, INPUT);
  Serial.begin(115000);
  interruptSetup();
  Serial.println("CLEARDATA");
  Serial.println("LABEL,Time,val,ROW");
}

void interruptSetup(){ //seta Timer2 para gerar interrupção every
other ms.
  TCCR2A = 0x02; //assim se tem sample rate de 500 Hz
  TCCR2B = 0x06; //desabilita PWM no pino 3 e 11 e o comando
tone()
  OCR2A = 0x7C; // timer2 em modo CTC, conta até 124 com
Prescaler de 124 (gasta 2ms para contar até 124).
```

```

    TIMSK2 = 0x02; //sempre q chega em 124, interrupt flag é
setada e a função Interrupt Service Routine (ISR) é rodada.
    sei(); //garante que interrupções globais
estão ativadas.
}

ISR(TIMER2_COMPA_vect){
    //função chamada a cada 2 ms para ler sensor
    Signal = analogRead(pulsePin);
    // le sensor em porta analógica
    sampleCounter += 2;
    //incrementa variável sampleCounter para acompanhar o tempo
    int N = sampleCounter - lastBeatTime;
    //variável N será usada para evitar ruídos mais tarde
    if(Signal < thresh && N > (IBI/5)*3){
        //Para evitar ruídos e leituras falsas, sinal deve ser >512(metade
do total analogico e N deve ter passado 5/3 do valor de IBI
        if (Signal < T){
            //Se sinal for menor que T (512)
            T = Signal;
            // faz T = Sinal, alterando valor do through
        }
    }
    if(Signal > thresh && Signal > P){
        //Se sinal > thresh e maior que
valor de pico (512)
        P = Signal;
        //atualize valor de
pico para o do sinal
    }
    //VERIFICANDO se
há pulso
    if (N > 250){
        //Aguarde 250 ms para
evitar ruídos de alta frequencia, assim seta limite superior em 240 bpm
        if ((Signal > thresh) && (Pulse == false) && (N > ((IBI/5)*3))){
            //se forma de onda passa valor de thresh (512) e 3/5 de
IBI passaram
            Pulse = true;
            //Ha pulso. Seta Pulse flag
            IBI = sampleCounter - lastBeatTime;
            //calcula tempo desde o ultimo beat para ter o
IBI
            lastBeatTime= SampleCounter;
            //atualiza lastBeatTime para sampleCounter
que iniciou em zero e foi incrementado de 2 em 2 para contar tempo
            //Garantir valor inicial de BPM realistico

            if(secondBeat){
                // As condicionais de secondBeat e firstBeat (inicializadas como true
e false) fazem sair da ISR (rotina de interrupção com return());
                secondBeat = false;
                // Para descartar
primeiro valor de IBI, ruidoso
                for(int i=0; i<=9;
i++){ //BPM é tirado de uma
média dos ultimos 10 valores de IBI

```



```

        rate[i] = IBI;
        //Na segunda vez, IBI + correto.
Usado como seed para a array rate para ter BPM mais preciso;
    }
}
if(firstBeat){
    firstBeat = false;
    secondBeat = true;
    sei();
    return;
}
//CALCULANDO BPM
word runningTotal = 0; //variável grande
usada para coletar IBIs.
for(int i=0; i<=8; i++){
    rate[i] = rate[i+1];
    runningTotal += rate[i]; //contents de rate são
passados e adicionados em runningTotal
}
rate[9] = IBI; //IBI mais
antigo sai da posição 0
runningTotal += rate[9]; //IBI mais recente vai
para posição 9
runningTotal /= 10; //runningTotal =
runningTotal/10
BPM = 60000/runningTotal; //Tira-se média de IBIs
para calcular BPM
QS = true; //seta flag Quantified
Self
}
}
//DEFINIR NÃO BEATS
if (Signal < thresh && Pulse == true){ //Quando sinal passa pelo
valor tresh descendo, pulso está encerrado
    Pulse = false; //Pulso encerrado
    amp = P - T; //amplitude da ultima onda é
medida
    thresh = amp/2 + T; //tresh é atualizada com a
nova marca de 50%
    P = thresh; //P e T são atualizadas com o
novo valor de thresh
    T = thresh;
}
if (N > 2500){ // Caso Não haja beats por 2,5s
    thresh = 510; //variaveis são reiniciadas para valor
inicial
    P = 510;
    T = 510;
    firstBeat = true;
    secondBeat = false;
    lastBeatTime = sampleCounter; //Timer2 permite que o algoritmo
de achar beats esteja sempre ativo no background, subindo novos valores.
}
}
}

void loop(){
    // Lê a temperatura e converte para °C
    sendCommandSHT(comandoTemperaturamperatura, dataPin, clockPin);
    waitForResultSHT(dataPin);

```

```

    valT = getData16SHT(dataPin, clockPin);
    skipCrcSHT(dataPin, clockPin);
    temperatura = (float)valT * 0.01 - 40;
    // Lê a umidade
    sendCommandSHT(comandoUmidade, dataPin, clockPin);
    waitForResultSHT(dataPin);
    valH = getData16SHT(dataPin, clockPin);
    skipCrcSHT(dataPin, clockPin);
    umidade = -4.0 + 0.0405 * valH + -0.0000028 * valH * valH;
    //Manda dados para o Excel
    sendDataToExcel();
    if (QS == true){
        sendDataToProcessing('B',BPM);
        sendDataToProcessing('Q',IBI);
        fadeVal = 255;
        QS = false;
    }
    delay(1500); //4 amostras por minuto com 15000, 2 amostras por
minuto; 3000 USADO
}

void sendDataToExcel(){
    val = IBI;
    ROW++;
    Serial.print("IBI,TEMP,HUMIDTIME,TIME");
    Serial.print(val);
    Serial.print(",");
    Serial.print(valT);
    Serial.print(",");
    Serial.print(valU);
    Serial.print(",");
    Serial.print(ROW);
}

void sendCommandSHT(int command, int dataPin, int clockPin)
{
    int ack;
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    digitalWrite(dataPin, HIGH);
    digitalWrite(clockPin, HIGH);
    digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, LOW);
    digitalWrite(clockPin, HIGH);
    digitalWrite(dataPin, HIGH);
    digitalWrite(clockPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, command);
    digitalWrite(clockPin, HIGH);
    pinMode(dataPin, INPUT);
    ack = digitalRead(dataPin);
    if (ack != LOW)
        Serial.println("ACK error 0");
    digitalWrite(clockPin, LOW);
    ack = digitalRead(dataPin);
    if (ack != HIGH)
        Serial.println("ACK error 1");
}

```

```

void waitForResultsHT(int dataPin)
{
    int ack;
    pinMode(dataPin, INPUT);
    for (int i = 0; i < 100; ++i)
    {
        delay(20);
        ack = digitalRead(dataPin);
        if (ack == LOW)
            break;
    }
    if (ack == HIGH)
        Serial.println("ACK error 2");
}

int getData16SHT(int dataPin, int clockPin)
{
    int val;
    pinMode(dataPin, INPUT);
    pinMode(clockPin, OUTPUT);
    val = shiftIn(dataPin, clockPin, MSBFIRST, 8, 3);
    val *= 256;
    pinMode(dataPin, OUTPUT);
    digitalWrite(dataPin, HIGH);
    digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
    pinMode(dataPin, INPUT);
    val |= shiftIn(dataPin, clockPin, MSBFIRST, 8, 3);
    return val;
}

void skipCrcSHT(int dataPin, int clockPin)
{
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    digitalWrite(dataPin, HIGH);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
}

```

## ANEXO B – Código de ativação do sistema de controle

```
const int LED = 15;
const int BUZZER = 3;
const int POT = 2;
const int pulsePin = 35;
const int clockPin = 34; // pin para clock
const int dataPin = 33; // pin para dado
int ROW = 0;
int LABEL = 1;
int val = 0;
int comandoTemperatura = B00000011; // comando usado para ler temperatura
int comandoUmidade = B00000101; // comando usado para ler umidade
int ack; // variavel para erros
int valT;
int valU;
float temperatura;
float humidade;
volatile int BPM; //Declaradas como volatile por serem usadas
no ISR e em outras partes do codigo
volatile int Signal;
volatile int IBI = 600;
volatile boolean Pulse = false;
volatile boolean QS = false;
volatile int rate[10];
volatile unsigned long sampleCounter = 0;
volatile unsigned long lastBeatTime = 0;
volatile int P = 80;
//valor de pico
volatile int T = 80;
//valor through
volatile int thresh = 80; //tresh
inicializado em 512 ms (metade da amplitude analogica)
volatile int amp = 100;
volatile boolean firstBeat = true;
volatile boolean secondBeat = false;

void setup(){
    pinMode(LED, OUTPUT);
    pinMode(Buzzer, OUTPUT);
    pinMode(POT, INPUT); // Por padrão porta analógica já é
INPUT
    pinMode(pulsePin, INPUT);
    pinMode(clockPin, INPUT);
    pinMode(dataPin, INPUT);
    Serial.begin(115000);
    interruptSetup();
    Serial.println("CLEARDATA");
    Serial.println("LABEL,Time,val,ROW");
}

void interruptSetup(){ //seta Timer2 para gerar interrupção every
other ms.
    TCCR2A = 0x02; //assim se tem sample rate de 500 Hz
    TCCR2B = 0x06; //desabilita PWM no pino 3 e 11 e o comando
tone()
    OCR2A = 0x7C; // timer2 em modo CTC, conta até 124 com
Prescaler de 124 (gasta 2ms para contar até 124).
```

```

    TIMSK2 = 0x02; //sempre q chega em 124, interrupt flag é
setada e a função Interrupt Service Routine (ISR) é rodada.
    sei(); //garante que interrupções globais
estão ativadas.
}

ISR(TIMER2_COMPA_vect){
    //função chamada a cada 2 ms para ler sensor
    Signal = analogRead(pulsePin);
    // le sensor em porta analógica
    sampleCounter += 2;
    //incrementa variável sampleCounter para acompanhar o tempo
    int N = sampleCounter - lastBeatTime;
    //variável N será usada para evitar ruídos mais tarde
    if(Signal < thresh && N > (IBI/5)*3){
        //Para evitar ruídos e leituras falsas, sinal deve ser >512(metade
do total analogico e N deve ter passado 5/3 do valor de IBI
        if (Signal < T){
            //Se sinal for menor que T (512)
            T = Signal;
            // faz T = Sinal, alterando valor do through
        }
    }
    if(Signal > thresh && Signal > P){
        //Se sinal > thresh e maior que
valor de pico (512)
        P = Signal;
        //atualize valor de
pico para o do sinal
    }
    //VERIFICANDO se
há pulso
    if (N > 250){
        //Aguarde 250 ms para
evitar ruídos de alta frequencia, assim seta limite superior em 240 bpm
        if ((Signal > thresh) && (Pulse == false) && (N > ((IBI/5)*3))){
            //se forma de onda passa valor de thresh (512) e 3/5 de
IBI passaram
            Pulse = true;
            //Ha pulso. Seta Pulse flag
            IBI = sampleCounter - lastBeatTime;
            //calcula tempo desde o ultimo beat para ter o
IBI
            lastBeatTime= SampleCounter;
            //atualiza lastBeatTime para sampleCounter
que iniciou em zero e foi incrementado de 2 em 2 para contar tempo
            //Garantir valor inicial de BPM realistico

            if(secondBeat){
                // As condicionais de secondBeat e firstBeat (inicializadas como true
e false) fazem sair da ISR (rotina de interrupção com return());
                secondBeat = false;
                // Para descartar
primeiro valor de IBI, ruidoso
                for(int i=0; i<=9;
i++){ //BPM é tirado de uma
média dos ultimos 10 valores de IBI

```

```

        rate[i] = IBI;
        //Na segunda vez, IBI + correto.
Usado como seed para a array rate para ter BPM mais preciso;
    }
}
if(firstBeat){
    firstBeat = false;
    secondBeat = true;
    sei();
    return;
}
//CALCULANDO BPM
word runningTotal = 0; //variável grande
usada para coletar IBIs.
for(int i=0; i<=8; i++){
    rate[i] = rate[i+1];
    runningTotal += rate[i]; //contents de rate são
passados e adicionados em runningTotal
}
rate[9] = IBI; //IBI mais
antigo sai da posição 0
runningTotal += rate[9]; //IBI mais recente vai
para posição 9
runningTotal /= 10; //runningTotal =
runningTotal/10
BPM = 60000/runningTotal; //Tira-se média de IBIs
para calcular BPM
QS = true; //seta flag Quantified
Self
}
}
//DEFINIR NÃO BEATS
if (Signal < thresh && Pulse == true){ //Quando sinal passa pelo
valor tresh descendo, pulso está encerrado //Pulso encerrado
    Pulse = false; //amplitude da ultima onda é
    amp = P - T; //tresh é atualizada com a
medida //nova marca de 50%
    thresh = amp/2 + T; //P e T são atualizadas com o
    P = thresh;
    novo valor de thresh
    T = thresh;
}
if (N > 2500){ // Caso Não haja beats por 2,5s
    thresh = 510; //variaveis são reiniciadas para valor
inicial
    P = 510;
    T = 510;
    firstBeat = true;
    secondBeat = false;
    lastBeatTime = sampleCounter; //Timer2 permite que o algoritmo
de achar beats esteja sempre ativo no background, subindo novos valores.
}
}
}

void loop(){
    // Lê a temperatura e converte para °C
    sendCommandSHT(comandoTemperaturamperatura, dataPin, clockPin);
    waitForResultSHT(dataPin);

```

```

    valT = getData16SHT(dataPin, clockPin);
    skipCrcSHT(dataPin, clockPin);
    temperatura = (float)valT * 0.01 - 40;
    // Lê a umidade
    sendCommandSHT(comandoUmidade, dataPin, clockPin);
    waitForResultSHT(dataPin);
    valH = getData16SHT(dataPin, clockPin);
    skipCrcSHT(dataPin, clockPin);
    umidade = -4.0 + 0.0405 * valH + -0.0000028 * valH * valH;
    //Manda dados para o Excel
    sendDataToExcel();
    if (QS == true){
        sendDataToProcessing('B',BPM);
        sendDataToProcessing('Q',IBI);
        fadeVal = 255;
        QS = false;
    }
    delay(1500); //4 amostras por minuto com 15000, 2 amostras por
minuto; 3000 USADO
}

void sendDataToExcel(){
    val = IBI;
    ROW++;
    Serial.print("IBI,TEMP,HUMIDTIME,TIME");
    Serial.print(val);
    Serial.print(",");
    Serial.print(valT);
    Serial.print(",");
    Serial.print(valU);
    Serial.print(",");
    Serial.print(ROW);
}

void sendCommandSHT(int command, int dataPin, int clockPin)
{
    int ack;
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    digitalWrite(dataPin, HIGH);
    digitalWrite(clockPin, HIGH);
    digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, LOW);
    digitalWrite(clockPin, HIGH);
    digitalWrite(dataPin, HIGH);
    digitalWrite(clockPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, command);
    digitalWrite(clockPin, HIGH);
    pinMode(dataPin, INPUT);
    ack = digitalRead(dataPin);
    if (ack != LOW)
        Serial.println("ACK error 0");
    digitalWrite(clockPin, LOW);
    ack = digitalRead(dataPin);
    if (ack != HIGH)
        Serial.println("ACK error 1");
}

```

```

void waitForResultsHT(int dataPin)
{
    int ack;
    pinMode(dataPin, INPUT);
    for (int i = 0; i < 100; ++i)
    {
        delay(20);
        ack = digitalRead(dataPin);
        if (ack == LOW)
            break;
    }
    if (ack == HIGH)
        Serial.println("ACK error 2");
}

int getData16SHT(int dataPin, int clockPin)
{
    int val;
    pinMode(dataPin, INPUT);
    pinMode(clockPin, OUTPUT);
    val = shiftIn(dataPin, clockPin, MSBFIRST, 8, 3);
    val *= 256;
    pinMode(dataPin, OUTPUT);
    digitalWrite(dataPin, HIGH);
    digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
    pinMode(dataPin, INPUT);
    val |= shiftIn(dataPin, clockPin, MSBFIRST, 8, 3);
    return val;
}

void skipCrcSHT(int dataPin, int clockPin)
{
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    digitalWrite(dataPin, HIGH);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
}

```