

UNIVERSIDADE FEDERAL DO PARANÁ

IGOR SIMÃO

LUCAS CAVALCANTE MARCELO

**SISTEMA DE SENSORES E COMUNICAÇÃO SEM FIO PARA MONITORAMENTO
DE SONO E AUTO QUANTIFICAÇÃO**

CURITIBA

2019

**IGOR SIMÃO
LUCAS CAVALCANTE MARCELO**

**SISTEMA DE SENSORES E COMUNICAÇÃO SEM FIO PARA MONITORAMENTO
DE SONO E AUTO QUANTIFICAÇÃO**

Monografia apresentada à disciplina “Projeto de Graduação - TE105” como requisito parcial à conclusão do Curso de Engenharia Elétrica, Setor Ciência e Tecnologia, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Eduardo Parente Ribeiro

**CURITIBA
2019**

*Dedico este trabalho aos que defendem a
educação e lutam por uma Universidade Pública
gratuita para todas e todos.*

Lucas Cavalcante Marcelo

*Dedico este trabalho aos estudantes de
Engenharia Elétrica da UFPR que, apesar
dos desafios impostos durante o caminho,
se ajudam, trabalham duro, e os vencem.*

Igor Simão

AGRADECIMENTOS

Eu, Lucas Cavalcante Marcelo, agradeço primeiramente à minha esposa Mariana Mielke pelas incontáveis vezes nas quais fora minha inspiração para arquitetar sonhos que se tornaram realidade, por ser minha casa onde quer que estivéssemos e por me ensinar humildemente a ser uma pessoa melhor, procurando minhas falhas dia a dia, para corrigi-las sempre a cuidar do próximo. De tanto não parar a gente chegou lá.

Também agradeço à minha família: aos meus pais pelo apoio, suporte por toda a vida e incentivo à educação; ao meu irmão que foi, e sempre será, minha grande referência; às minhas cunhadas, sogro e sogra por mostrarem com amor o real significado de Família. À Sheila, Luiz, Beatriz e Larissa e Zé pela hospitalidade, carinho e acolhimento nos momentos que precisei ao longo desta trajetória.

Aos meus amigos que acompanham desde 2011, pelos quais fui acolhido e aos quais jamais hesitarei acolher. Ao Igor, cuja humildade e hospitalidade, que mesmo depois de 7 anos, ainda me causam admiração, tamanha é a grandeza. Ao Leandro, que me ajudou diversas vezes com os códigos em Java, sempre de maneira cuidadosa e ágil, sem as quais e talvez ainda estivesse procurando por uma solução. Ao Rafael por ceder seu monitor para que eu pudesse trabalhar mais confortavelmente.

Ao Professor Dr. Eduardo Parente Ribeiro, pelo conhecimento, impossível de dimensionar, adquiridos nas aulas, corredores e gabinete, sempre com muito respeito e carinho. Também pela motivação que nos deu, trazendo mais sentido à esta graduação cuja beneficência, apesar de todo seu esforço, pertence somente a nós alunos. Também por todas e todos que lutam em defesa da educação e da Universidade Pública gratuitos para toda a população brasileira.

A todos vocês, muito obrigado.

Eu, Igor Simão, agradeço primeiramente à minha família por toda paciência e investimento que aplicaram em mim. Em especial aos meus dois irmãos Ricardo e Leandro nos quais sempre me motivaram a não abandonar o barco e continuar sempre remando, seja para a frente ou para trás – mas sempre remando. Ao meu pai e minha mãe que além dos sacrifícios já feitos estão sempre dispostos a mais alguns sem pensar duas vezes. Ao Luis Alberto e sua família, que nunca postergaram

atenção e conselhos que me sustentam até hoje, sou muito grato. A minha namorada e também melhor amiga, Natasha, que sempre esteve ao meu lado durante a graduação com carinho e participação nas minhas dificuldades. Agora não teremos mais desculpas para deixar de assistir O Senhor dos Anéis.

Agradecimento também a todo pessoal da elétrica, em especial ao Leonardo que sempre esteve disponível compartilhando paçocas e sabedorias mineiras, nunca deixou de ser um grande amigo mesmo longe. Ao Lucas Cavalcante que teve muita paciência comigo e soube lidar com qualquer tipo de instabilidade – até mesmo sinais ruidosos nos nossos sensores de batimento cardíaco. Ao professor Dr. Parente que me adotou durante o curso sendo um excelente profissional que tive prazer em trabalhar por dois anos com sensoriamento. Com ele aprendi muito sobre instrumentação eletrônica e análise de sinais – principais fundamentos deste projeto de conclusão de curso.

Ao Marcelo Abrantes, Thomas, Matt e Hector Gil que me adotaram numa terra forasteira cheia de desafios, e me ensinaram a extrair experiência e aproveitar as oportunidades dos piores momentos que surgem na vida. Pois são nessas ocasiões intensas que o autoconhecimento é mais valioso, e a experiência é maior.

*o tempo
entre o sopra
e o apagar da vela*

Paulo Leminski

RESUMO

Por ser um tema atual no mercado e com muita tendência de crescimento, a auto quantificação está sendo alvo de aplicações para diversas áreas de estudo, pois através de estatísticas podem se determinar padrões e correlações entre um sistema biológico e comportamental. Um exemplo disso são os relógios inteligentes que fazem coleta do batimento cardíaco (sistema biológico) e o correlacionam com atividade física de uma pessoa (sistema comportamental), proporcionando ao usuário um autoconhecimento de sua performance física. A auto quantificação neste projeto se aplica no monitoramento de sono fazendo coleta de sinais vitais relacionados à qualidade do sono do usuário. Após a coleta concluída, várias análises comportamentais desses sinais podem ser extraídas a partir dos resultados. Destes possíveis sinais vitais, foram implementados sensores de frequência respiratória, para verificar variações e possíveis casos de apneia, batimento cardíaco, sensores de movimentação noturna, para apontar horas de sono profundo ou instabilidade do sono, bruxismo e temperatura. Estes sensores, juntos, fornecem dados para estudo da a qualidade do sono do usuário através da análise da correlação entre os sinais durante todo período de coleta. Para isso, foi necessário implementar todos esses sensores e adequá-los para coletar os dados vitais de uma forma não intrusiva - principalmente aplicando protocolos de comunicação sem fio. Um aplicativo *Android* foi desenvolvido, cujo objetivo é receber os valores lidos pelos sensores via *bluetooth* (operando no modo de baixo consumo) e mostrá-los em forma de gráficos e tabelas através da Interface Gráfica do Usuário. Dessa forma, o usuário pode analisar seus próprios dados vitais do período de descanso noturno.

Palavras-chave: Análise do sono. *Android*. Auto quantificação.

ABSTRACT

For being current in the market and a rising trend, the quantified self is being a target for several applications and study areas, for numeric statistics may be used to conclude patterns and correlations between a biological and behavior systems. The smartwatch is an example of that, it collects the heart beating frequency (biological system) and relate it with the personal physical activity (behavior system), allowing a self knowledge of your own athletic performance. The quantified self of this project applies to the sleep monitoring to collect the body's vital signals and relate with the sleep quality. After the signal collecting is done, several behavior analysis might be released from the collected results. Those possible vital signals, the focus will be the respiratory frequency to verify the frequency variations and possible apnea cases (when the user stops breathing during a short night period), heart beating, night moving drive sensor to appoint deep sleep hours during the night, or sleep instability. Also the development of teeth grinding sensor and temperature collecting. Those sensors together will support the sleep quality analysis during all the collecting period of the night, and for that will be necessary to develop them all and set best adequation for collect data using the less intrusive actions - giving away from physical communications to handling wireless communication protocols. To interact the user with the collected data, following the principles of Quantified Self, an *Android* app was developed, which the goal is to receive the sensors values by *Bluetooth* Low Energy and display them on a Graphical User Interface in the form of graphs and tables. In this way, the user might conclude data analyses from the sleep time during the night.

Key-words: Sleep analysis. *Android*. Quantified Self.

LISTA DE ILUSTRAÇÕES

Figura 1 – Resultado da pesquisa realizada pelo IPOM.....	13
Tabela 1 - Resultado sobre estudo de caso sobre Estresse e Qualidade de Sono...	14
Figura 2 - Topologia do projeto abrangendo os sensores que serão implementados e os meios que resumem a comunicação dos sensores ao celular <i>Android</i>	16
Figura 3 - Mapa da cidade de Portland com rotas percorridas.....	18
Figura 4 - Maratona de Curitiba registrada em 2018.....	18
Figura 5 - Equipamentos disponíveis no mercado.	19
Figura 6 - Variação da frequência cardíaca durante atividade física.	20
Figura 8 – Oscilações da respiração natural. Uma media de doze ciclos por minuto que alternam entre inspiração e expiração.....	22
Figura 10 – Módulo de desenvolvimento ESP32 DEV KIT V1.	23
Tabela 2 - Sistemas operacionais dos <i>smartphones</i> adquiridos no Brasil em 2017..	24
Figura 11 - Ensaio realizado para isolamento de ondas eletromagnéticas.	26
Figura 12 – Ilustração dos testes feitos em laboratório para validação do sistema óptico de reflexão da luz e sua proporcionalidade à concentrações de tinta.....	27
Figura 13 – Tabela e resultados dos sinais de tensão lidos para diferentes amostras de tinta.....	27
Figura 14 – Resultado da variação luminosa visto na tela do osciloscópio.	28
Figura 15 - Sensor óptico à esquerda usado para detecção de batimento cardíaco e microcontrolador arduíno à direita usado para fazer coleta das variações luminosas recebidas pelo fotorreceptor do sensor óptico.....	28
Figura 16 – Sinal fisiológico do batimento cardíaco de uma pessoa.	29
Figura 17 - Intensidade luminosa variando conforme os batimentos cardíacos.	29
Tabela 3 – Valores analógicos proporcionais à intensidade luminosa lida como amostras armazenados em sessenta índices do vetor.	30
Figura 18 - O algoritmo armazena a sequência onde o elemento AD pela primeira vez fica maior que o valor médio (T1). Quando é registrado a segunda vez (T2), o algoritmo encerra seu cálculo fazendo a diferença entre as sequências T1 e T2.	30
Figura 19 – Diagrama mostrando as etapas que o algoritmo calcula o batimento cardíaco médio.....	31
Figura 20 - O gráfico acima mostra uma aquisição de dados ao longo de nove minutos de uma coleta noturna.	32

Figura 21 – Cinta torácica onde foi anexada o invólucro de ar que contém o sensor de pressão BMP180.....	33
Figura 22 - Fluxograma do algoritmo disparado por uma chave mecânica para validar uma coleta noturna.	33
Figura 23 – Resultado da variação da pressão interna do invólucro de ar da cinta torácica.....	34
Figura 24 - incidente recorrente que causa inversões duplas na derivada da reta tangente na mesma crista do sinal.	35
Figura 25 - Resultado de uma coleta curta para validar o sensor de frequência respiratória. A interferência gerada no sinal pelo ruído vem de movimentos físicos enquanto a coleta estava sendo feita.	36
Tabela 4 - Resultado das frequências respiratórias lidas do autor e sua média no minuto completo. Durante o intervalo de teste, manualmente, foram contabilizados dezesseis ciclos completos bem como a média mostrada na tabela.	36
Figura 26 - Imagem do sensor de bruxismo. Formado por duas camadas de silicone moldadas com a arcada dentária do usuário e que englobam o sensor de pressão digital. Nela há quatro fios de funcionamento e comunicação dos dados.	37
Figura 27 – Simulação da ocorrência de bruxismo captada pelo sensor desenvolvido..	37
Figura 28 – Funcionamento do algoritmo para detecção de movimentos noturnos. .	38
Figura 29 – Resultado de uma coleta noturna realizada nos autores.	39
Figura 30 - Curva de resposta das intensidades atribuídas pelo sensor MPU6050 em resposta a um ângulo isolado.....	40
Figura 31 – Variação da temperatura coletada no interior do invólucro de ar do sensor de frequência respiratória. Houve um desacoplamento próximo à meia noite e meia e no período de duas e meia da manhã até as cinco horas	41
Figura 32 - Estrutura do perfil do GATT.	44
Figura 33 - Algoritmo para comunicação entre o ESP32 e o Android.	45
Figura 34 - Estrutura do módulo do aplicativo.	46
Tabela 5- Tabela retirada do manual do ESP32 referente a consumacao proporcional aos seus estados de transmissão sem fio.	52
Equação 1 - Equação para a capacidade estimada da bateria.	53
Figura 35 - Imagem da bateria lítio-polímero de 1800mAh usada para suportar o processamento do ESP com a transmissão de dados.	53

Figura 36 - Esquemático mostrando o conversor AD coletando a tensão de descarga da bateria.	54
Figura 37 - Monitoramento da descarga da bateria do sistema de transmissão e processamento de dados do ESP32.	54
Figura 38 - Resultado do monitoramento noturno do batimento cardíaco.....	56
Figura 39 - Resultado do monitoramento noturno da frequência respiratória.....	56
Tabela 6 - Período de sono profundo detectado pelo sensor de movimentação noturna.	57
Figura 40 - Acompanhamento da temperatura lida no interior do invólucro de ar da cinta torácica em paralelo com a respiração média..	58
Figura 41 – Resultado da relação da frequência respiratória com os batimentos cardíacos médios no período de sono profundo e REM.....	58
Figura 42 - Atividade do aplicativo exibindo os resultados de uma coleta de teste.	60
Figura 43 – Tabela de características do protótipo.	61

SUMÁRIO

LISTA DE ILUSTRAÇÕES	8
1 INTRODUÇÃO	12
1.1 OBJETIVO GERAL.....	15
1.2 OBJETIVOS ESPECÍFICOS.....	15
1.3 ESTRUTURA.....	16
2 FUNDAMENTAÇÃO	17
2.1 AUTO QUANTIFICAÇÃO E ENGENHARIA	17
2.2 SINAIS VITAIS RELACIONADOS AO SONO.....	20
2.2.1 Batimento Cardíaco Médio.....	20
2.2.2 Bruxismo	21
2.2.3 Frequência Respiratória.....	22
2.3 MICROCONTROLADOR E MÓDULO <i>BLUETOOTH</i>	23
2.4 <i>SMARTPHONE</i> E SISTEMA OPERACIONAL.....	23
3 DESENVOLVIMENTO.....	25
3.1 ELIMINANDO NÍVEIS SIGNIFICATIVOS DE RUÍDO USANDO A MALHA DOS CABOS DE COMUNICAÇÃO.....	25
3.2 IMPLEMENTAÇÃO DO SENSOR DE BATIMENTO CARDÍACO.....	26
3.3 IMPLEMENTAÇÃO E CONFECÇÃO DO SENSOR DE FREQUÊNCIA RESPIRATÓRIA.....	32
3.4 IMPLEMENTANDO O SENSOR DE BRUXISMO.....	36
3.5 IMPLEMENTANDO O SENSOR DE MOVIMENTAÇÃO NOTURNA.....	38
3.6 CALIBRAÇÃO DAS INTENSIDADES DE MOVIMENTAÇÃO NOTURNA....	40
3.7 APLICAÇÃO DO SENSOR DE TEMPERATURA PARA VALIDAÇÃO DOS SINAIS VITAIS	41
3.8 APLICATIVO <i>ANDROID</i>	42
3.8.1 Editor e outras ferramentas	42
3.8.2 Comunicação com o <i>Smartphone</i>.....	43
3.8.3 Protocolo de comunicação	44
3.9 ESTRUTURA DO CÓDIGO	46
3.9.1 Módulo	46
3.9.2 Manifesto.....	47
3.9.2.1 Permissões.....	47

3.9.2.2	Recursos de <i>hardware</i> e <i>software</i> necessários para aplicativo	47
3.9.2.3	Atividades	48
3.9.2.4	Serviços	48
3.9.3	Recursos	48
3.10	CÓDIGO DA APLICAÇÃO <i>ANDROID</i> EM JAVA	49
3.10.1	Atividade de busca de dispositivos próximos	49
3.10.2	Atividade de controle de dispositivo conectado	50
3.10.3	Serviço de controle da comunicação <i>bluetooth</i>	51
3.11	VERIFICAÇÃO DO CONSUMO DO SISTEMA E BATERIA.....	52
4	RESULTADOS	55
4.1	RESULTADO DAS COLETAS DOS SENSORES E CORRELAÇÕES	55
4.2	INTERFACE GRÁFICA DO APLICATIVO	59
4.3	DISCUSSÃO	60
5	CONCLUSÃO.....	62
5.1	TRABALHOS FUTUROS.....	63
	REFERÊNCIAS.....	65

1 INTRODUÇÃO

Uma rede de sensores tem um vasto espectro de aplicações que se estende para diversas áreas como a indústria automobilística, automação residencial, monitoramento de dados ambientais e auto quantificação, descrita no capítulo 2.2. Na indústria automobilística, por exemplo, os sensores são aplicados nos freios com *Anti-lock Braking System* (ABS) no qual há o envio de sinais do sistema de frenagem do carro à unidade central para garantir mais segurança ao condutor quando ocorre uma frenagem repentina. Também, em paralelo ao freio ABS, são feitas leituras de emissão de combustível para melhorar a performance de explosão dos pistões. Estes exemplos mostram a aplicação dos sensores, desde a segurança do motorista até a performance do motor. Outro exemplo são as redes de sensores ambientais que fazem registros de temperatura, pressão atmosférica e velocidade do vento. Esses sinais são correlacionados para levantar informações sobre o comportamento do clima ao longo do dia.

Em uma rede de sensores, são definidas variáveis de entrada, meios de processamento e variáveis de saída. As variáveis de entrada são os sinais físicos lidos pelos sensores (temperatura, pressão atmosférica e velocidade do vento). Os meios de processamento são as tecnologias embarcadas responsáveis pela transformação e análise dos sinais. A variável de saída é o sinal, o resultado final do processamento. No caso deste trabalho, as variáveis de saída são frequências cardíaca e respiratória, temperatura, nível de movimentação e força mandibular indicativa de bruxismo. As variáveis de entrada são variações ótica do fluxo sanguíneo, variação do diâmetro do tórax, temperatura e variação da aceleração e pressão entre os dentes. Os meios de processamento são o microcontrolador e o telefone móvel, que fazem as leituras e processam os dados.

A auto quantificação visa, a partir de registros biológicos e comportamentais, possibilitar melhorias para controle tendo em vista que podem ser mostrados padrões que se relacionam entre os sinais registrados. Um exemplo disso são os relógios inteligentes voltados ao esporte como a corrida, que possuem sensores capazes de captar diversos sinais biológicos relacionados à performance atlética como: velocidade instantânea, cadência (número de passos por minuto), batimento cardíaco e altimetria do percurso. Por fim, esses dados são processados e oferecem ao usuário informações sobre ritmo e momentos de esforço físico (geralmente proporcional ao

batimento cardíaco). Assim o corredor pode entender o desempenho que ele teve em cada trecho do trajeto da atividade física. A principal correlação é o batimento cardíaco médio com a velocidade do atleta em paralelo com desnível altimétrico. Para isso, os relógios inteligentes devem ser ricos em sensores para extrair o máximo de informações possível a fim de disponibilizá-las ao atleta.

Em relação ao sono, cerca de 40% da população mundial tem algum problema com o sono (OMS, 2018). Um sono mal equilibrado, a curto prazo, pode provocar estresse, mal humor e afeta as atividades diárias. A longo prazo pode ser ainda mais grave, pois prejudica consideravelmente a saúde humana gerando problemas cardiovasculares, envelhecimento acelerado, obesidade e, em casos extremos, depressão e Acidente Vascular Cerebral (AVC) (CYMBALUK, 2019). Na Figura 1, os resultados da pesquisa realizada em 2012 com duas mil pessoas entrevistadas a respeito do que sentem quando passam por um sono insatisfatório relata que 82% das pessoas que tem um sono ruim ou insatisfatório sentem cansaço, sonolência e queda de rendimento durante o dia.

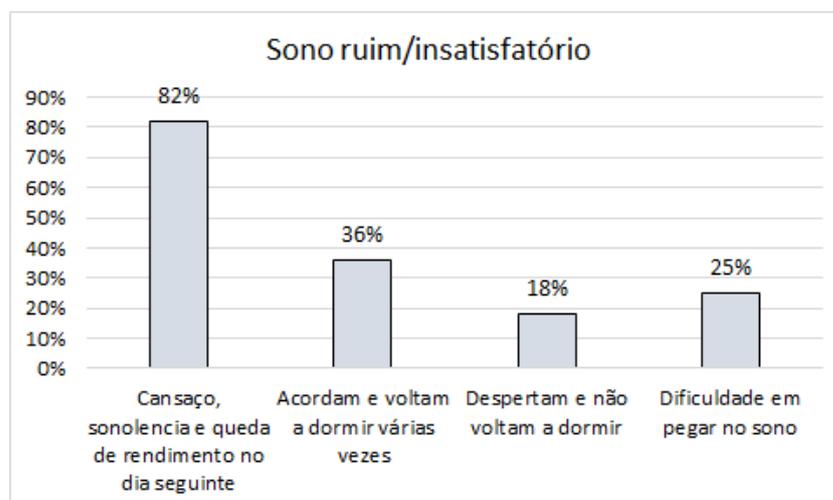


Figura 1 – Resultado da pesquisa realizada pelo IPOM.
Fonte: IPOM, 2012.

A consequência mais observada em um grupo de pessoas que não tem o sono adequado é o nível de estresse. Na Tabela 1 é mostrada relação do estresse com a qualidade de sono entre enfermeiras numa rotina hospitalar. Ela demonstra que 73% das enfermeiras apresentaram alto nível de estresse devido à má rotina de sono.

Tabela 1 - Resultado sobre estudo de caso sobre Estresse e Qualidade de Sono.
Fonte: ROCHA E MARTINO, 2010.

Comparação entre os níveis de stresse e qualidade de sono			
VARIÁVEL		(%) Bom sono	(%) Mal sono
< 2.0	Baixo stresse	100	0
2.0 a 2.9	Estresse médio	59	40,9
> 3.0	Alto stresse	26,67	73,33

Segundo a OMS, em 2018, cerca de 90% da população mundial é atingida pelo estresse. Apesar de ser natural do corpo humano como forma de estímulo para realizar algum trabalho e outros feitos (como TCC), em grandes níveis as consequências são diversas. No Brasil, 70% da população é atingida pelo estresse, e 30% estão com níveis altíssimos (OMS, 2019).

Desta forma, utilizando-se da auto quantificação, no monitoramento de sono deve haver sensores voltados à coleta de sinais vitais relacionados ao comportamento noturno do usuário. O sensor ótico de batimento cardíaco, como usado nos relógios e pulseiras inteligentes, obtém informações sobre as variações da frequência cardíaca durante todo período de monitoramento. O sensor de diâmetro do tórax permite a obtenção da frequência respiratória pelo acompanhamento dos períodos da inspiração e expiração a fim de reconhecer anomalias como a apneia. Em paralelo com estes sensores, também está acoplado o sensor de aceleração para detecção de movimentação noturna. Ele é responsável por detectar períodos de sono profundo (quantidade de horas que não há movimentação durante o sono), e quantas vezes o usuário se movimenta na cama. Há também o sensor de pressão dos dentes para indicação de bruxismo a fim de verificar com que frequência o usuário tem este tipo de desordem.

Os sensores utilizados neste trabalho necessitam, por natureza, ser lidos por um microcontrolador, assim como os sinais recebidos deles devem ser processados. Para isto, o ESP 32 foi considerado como uma boa opção por ter baixo consumo e capacidade de processamento de dados suficiente, ao mesmo tempo que contém adaptadores de comunicação sem fio (*Bluetooth Low Energy* e *Wi-fi*). Este tipo de comunicação é importante pois um dos objetivos da auto quantificação e, da mesma forma, deste trabalho, é exibir os dados coletados do usuário para ele mesmo. Tendo isto em vista, implementar um *display* ou qualquer outro tipo de interface com o usuário seria financeiramente custoso do ponto de vista do produto final. Portanto, a solução

encontrada foi a utilização de *smartphones*. Desta forma, o usuário final precisaria adquirir apenas o microcontrolador e os sensores.

Após avaliar as possíveis plataformas para utilizar no projeto, a interface gráfica foi definida como o método de exibição dos dados coletados durante a noite para o usuário, sendo que os dados lidos pelo microcontrolador deve ser enviados para o *smartphone* via *Bluetooth*.

1.1 OBJETIVO GERAL

O projeto tem como objetivo extrair sinais vitais relacionados do usuário durante o período noturno para fazer análises de suas variações e possíveis comportamentos que possam inferir na qualidade do sono. Planeja-se utilização do celular para armazenamento e visualização dos dados, uma plataforma popular e acessível, onde o usuário poderá avaliar o comportamento do próprio sono através de um aplicativo.

1.2 OBJETIVOS ESPECÍFICOS

Para alcançar os objetivos da auto quantificação, em coletar os sinais vitais e rearranjá-los em uma plataforma de fácil acesso, são necessários definir primeiramente quais são as variáveis de saída do sistema (variáveis que indicam qualidade do sono). Depois disso, definir as variáveis de entrada ou quais são os sensores que devem ser implementados para se obter as variáveis de saída. Por consequência os meios de processamento devem ter viabilidade e disponibilidade para adequá-los ao sistema que se deseja implementar. No caso, como a rede de sensores se aplica ao monitoramento de sono, motiva-se usar o mínimo de material intrusivo evitando-se fios e cabos, e direcionando a atenção para comunicação sem fio. Outro objetivo é utilizar o sistema operacional *Android* na rede de sensores, por ser de comumente disponível. Devem ser implementados sensores capazes de ler os sinais vitais relacionados ao sono como: respiração, batimento cardíaco, pressão mandibular, movimentação noturna e temperatura. Os sensores devem ser implementados separadamente, calibrados ou caracterizados, e bem condicionados ao corpo humano para um período de coleta.

Implementar uma plataforma de comunicação “sem fio”. Para isso será avaliado o uso do ESP32 que tem uma forte popularidade no mercado e apresenta todas as funções adequadas para o projeto como: comunicação *bluetooth*, microprocessador

embarcado e modos de baixo consumo que podem ser aplicados para uma coleta extensa de dados em operação com bateria.

Desenvolver aplicativo *Android* para disponibilizar ao usuário os resultados dos dados vitais lidos durante a noite. O *Android Studio* será a plataforma usada para desenvolver o aplicativo (GUI - *Graphical User Interface* ou Interface Gráfica com o Usuário, em português) que mostrará as características dos sinais vitais lidos.

Por fim, um dos resultados que se deseja obter é testar a integração dos sensores e fazendo análises noturnas e correlacionando todos os sinais lidos para verificar suas interações e padrões dos dados vitais.

A Figura 2 mostra a topologia a ser implementada com os equipamentos conectados ao corpo e comunicação bluetooth responsável pela transmissão dos dados dos sensores para o celular.

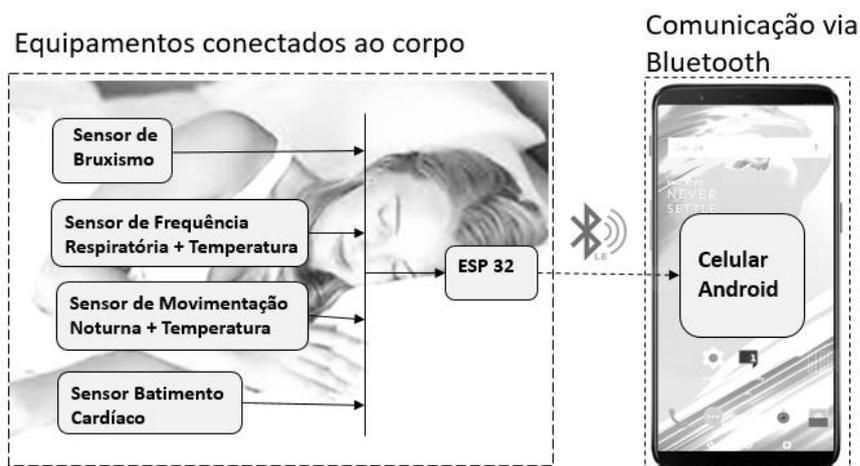


Figura 2 - Diagrama do projeto mostrando os sensores que serão implementados e a comunicação ao celular *Android*.

Fonte: Os Autores.

1.3 ESTRUTURA

No capítulo da fundamentação deste projeto, são levantados os temas relacionados ao sono, como consequências de uma noite bem ou mal dormida, juntamente às variáveis que caracterizam sua qualidade. Após a fundamentação, o capítulo de desenvolvimento abrange a instrumentação desenvolvida para detectar sinais relacionados ao sono, comunicação para transmissão desses sinais ao celular, assim como implementação do aplicativo *Android*. O capítulo Resultados mostra as correlações entre os sinais obtidos e o quanto pode ser importante o monitoramento e análise de sinais.

2 FUNDAMENTAÇÃO

Para permitir a análise da qualidade do sono é necessário levantar alguns parâmetros que são os sinais usados como variáveis de entrada do sistema, para isso serão implementados sensores para converter os sinais físicos vitais em sinais elétricos que servirão de entrada de processamento ao monitoramento do sono. Alguns sensores devem ser condicionados, ou seja, após captar sinais físicos através de um transdutor deve-se colocá-los em um nível de tensão ou corrente que pode ser lidos por um conversor analógico digital. Em muitos casos, cabe fazer a calibração e a caracterização dos sensores para relacionar os níveis elétricos com o valor físico lido. Inicialmente anseia-se buscar quais são os sinais que interessam ao projeto. Com o início do estudo, foram levantadas bibliografias com informações e dados a respeito da saúde do sono e quais as consequências que uma noite mal dormida poderia trazer às pessoas. Em seguida foi estudada a viabilidade da auto quantificação e como o meio acadêmico da engenharia poderia contribuir com este tema. Com isso, foram buscados meios de sensoriamento adequados ao projeto, e como coletar e condicionar os sinais captados do próprio usuário. Em seguida, foi necessário avaliar como operar toda a rede de sensores, transmitir e analisar as medidas.

2.1 AUTO QUANTIFICAÇÃO E ENGENHARIA

O termo Quantified Self (QS) ou Auto Quantificação teve sua origem com os editores da revista *Wired*, Gary Wolf e Kevin Kelley em 2007 (CALZAVARA, 2019), que futuramente fundaram o *Quantified Self Labs* cuja missão é “dar suporte a novas descobertas sobre nós mesmos e nossas comunidades...”. O co-fundador da IndieWebCamp Aaron Parecki, em 2013, publicou o resultado de seu trabalho sobre auto quantificação: após cinco anos registrando sua posição com um GPS, a cada um a seis segundos, foi possível exibir com detalhes o mapa da sua própria cidade -

Portland, Or, EUA. Na Figura 3 podem ser observados com detalhes cada quadra e rua da cidade de Portland se juntar todos os pontos registrados pelo GPS.



Figura 3 - Mapa da cidade de Portland com rotas percorridas.

Fonte: <https://aaronparecki.com/>

Na auto quantificação, a engenharia tem como ramo principal o uso da Internet das Coisas (IOT – *Internet of Things*) para ter o autoconhecimento do corpo humano através de sinais vitais. Um grande exemplo é sua aplicação em atividades físicas. Como exemplo, a Garmin, empresa americana de tecnologia, desenvolve relógios inteligentes destinados a quem pratica atividades físicas como corridas, ciclismo ou natação. O relógio tem vários sensores integrados com o Sistema de Posicionamento Global (GPS – do inglês *Global Positioning System*) que entregam ao atleta o batimento cardíaco médio, velocidade, cadência (número de passadas por minuto), e a altimetria em cada ponto no trajeto de uma maratona, entre outros dados. Na Figura 4 é mostrado o resultado dos sinais vitais do usuário registrados durante a maratona de Curitiba de 2018, onde permite ao maratonista avaliar os dados gráficos que relacionam o seu rendimento em cada ponto do percurso dos quarenta e dois quilômetros.

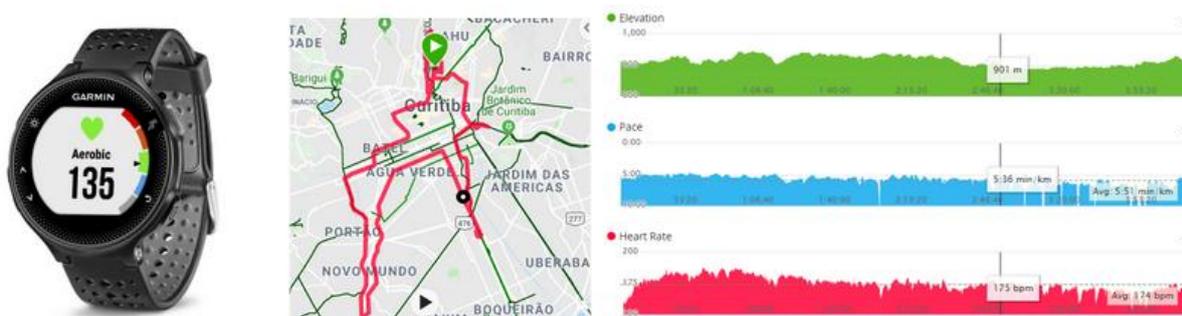


Figura 4 - Maratona de Curitiba registrada em 2018.

Fonte: Os Autores.

Já no projeto em desenvolvimento, a quantificação tem apoio no monitoramento de sinais característicos do usuário em seu horário de sono. Para isso será desenvolvido um grupo de sensores para extrair dados relevantes como frequência respiratória, batimento cardíaco, bruxismo e entre outros. Estes sensores serão agregados em uma rede de sensores que terá como objetivo mapear sinais a fim de que o usuário tenha um resultado em mãos para analisar seu comportamento durante a noite dormida.

Sabendo-se que há soluções no mercado para a auto quantificação do sono, foi realizando um levantamento de alguns equipamentos cujos objetivos são comuns a este trabalho. As características e funcionalidade de cada um dele foram descritas na Figura 5 para criar parâmetros de comparação a fim de guiar o desenvolvimento do protótipo.

Equipamento	Imagem	Sensores	Preço	Portabilidade	Conectividade	Confiabilidade
Phillips Efficia CM120 usado em hospitais para análises de sono.	 Fonte: cleanmedical.com.br	<ul style="list-style-type: none"> • 2x IBP (pressão arterial) • ECG • 2x Temperatura • SpO² • CO² • Entre outros 	Em torno de R\$ 80 mil	Instalado na maca	USB Ethernet	Possui tecnologia da Philips utilizada em mais de 200 milhões de pacientes por ano
Polissonígrafo Embletta Gold Portátil	 Fonte: mercadolivre.com.br	<ul style="list-style-type: none"> • SpO² • Fluxo de ar • Esforço torácico 	Em torno de R\$ 30 mil	Forma de pulseira (com fios até os sensores)	USB Ethernet	Compatível com as normas da AASM (American Academy of Sleep Medicine)
Xiaomi Mi Band 3 voltado ao público geral.	 Fonte: tecmundo.com.br	<ul style="list-style-type: none"> • Frequência cardíaca • Movimento 	Em torno de R\$ 100	Forma de pulseira (sem fio)	Bluetooth	Fabricante indica que os resultados são aproximados
POLYWATCH – Teste domiciliar de sono	 Fonte: bernafon.com.br	<ul style="list-style-type: none"> • SpO² • Frequência de Pulso • Fluxo de ar • Ronco • Movimento de pulso • Pressão de CPAP • Esforço torácico • Posição do corpo 	Em torno de R\$ 7 mil	Forma de pulseira (com fios até os sensores)	USB Bluetooth	Produto registrado na ANVISA sob o número 80117580477

Figura 5 - Equipamentos disponíveis no mercado.

2.2 SINAIS VITAIS RELACIONADOS AO SONO

2.2.1 Batimento Cardíaco Médio

O batimento cardíaco é um dos mais acessíveis indicadores que se pode ter a respeito do estado de saúde da pessoa, pois a frequência de batida (em situações normais) é proporcional ao esforço físico que ela está exercendo. Para um adulto normal, a média de batimentos está entre 60 e 100 batimentos por minuto (LIMA, 2019).

Usando um sensor de batimento cardíaco comercial oferecido pela Garmin para atividade de corrida, foi feito o seguinte experimento: cinco estímulos de 1000 m, onde a cada um quilômetro correndo, o autor descansava por um tempo de um minuto. Notou-se que os batimentos subiam de 75 bpm para 180 bpm apenas no intervalo de 1000 m correndo. A Figura 6 mostra os estímulos feitos pelo atleta sobrepostos com o batimento cardíaco médio. Conforme o esforço fica extremo, o batimento cardíaco o acompanha.

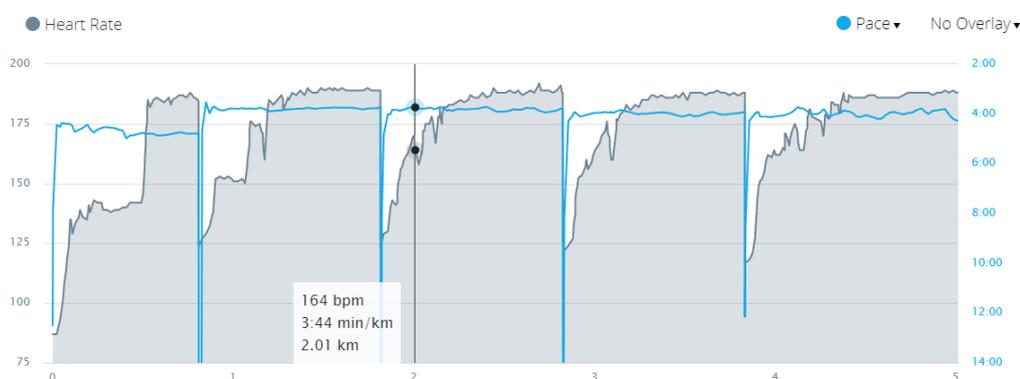


Figura 6 - Variação da frequência cardíaca durante atividade física.
Fonte: Os Autores.

Existem diversos modos de registrar ou medir o batimento cardíaco. O mais simples é o método manual, sentindo através do contato com o pulso quantas batidas o coração dá em um minuto. Em laboratórios ou hospitais são usadas conexões com eletrocardiógrafo através de eletrodos, já em áreas cirúrgicas são utilizados métodos restritos e intrusivos, diretamente na veia do paciente (apenas casos extremos onde não há força na pulsação devido ao grave estado de saúde).

A Figura 7 mostra as conexões do corpo com eletrocardiógrafo em cada membro do paciente para registro das derivações torácicas.

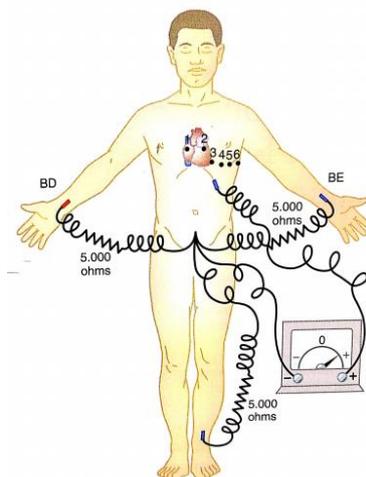


Figura 7 - Esquemático do eletrocardiógrafo conectado ao corpo humano.
Fonte: GUYTON e HALL, 2006.

O método óptico será utilizado e avaliado para o monitoramento durante o sono. Este sistema é simples e não tão intrusivo como o eletrocardiógrafo. Trata-se de aproveitar o fenômeno físico da reflexão da luz que tem intensidade refletida conforme o fluxo da corrente sanguínea. O funcionamento consiste em um emissor de luz e um fotorreceptor alinhados com a ponta do dedo, por exemplo. Conforme o coração bate, há um leve aumento no volume dos vasos sanguíneos do dedo fazendo que boa parte da luz, que seria recebida pelo receptor luminoso, seja absorvida pelo sangue. Este método é utilizado em esteiras de corrida, relógios inteligentes desportivos como os usados pelos autores, ou leitos hospitalares.

2.2.2 Bruxismo

O bruxismo é uma “desordem funcional provavelmente relacionada a fatores genéticos, situações de estresse, tensão, ansiedade ...” (BRUNA, 2019), e consiste no ranger dos dentes. A pressão da mandíbula causada pelo bruxismo pode causar desgaste e amolecimento entre os dentes. Em casos mais graves podem ocorrer problemas ósseos e gengivais (BRUNA, 2019).

Para obtenção dos sinais gerados pelo bruxismo, será desenvolvido um sensor para captar níveis de intensidades da mordida que o usuário aplica entre os dentes durante a noite sono.

2.2.3 Frequência Respiratória

A frequência respiratória tem como objetivo acompanhar quantas vezes o usuário inspirou e expirou num determinado tempo. Um adulto em repouso tem uma frequência respiratória média de doze incursões por minuto (GUYTON e HALL, 2006), e sua alteração pode “ocorrer em casos de falta de ar, ansiedade, choque, febre, etc.”. Ela pode ser regular ou irregular, não há um certo padrão, pois muda conforme a situação do momento. Durante o sono, um caso preocupante é a parada desta (apneia do sono). A Figura 8 mostra as variações de inspiração e expiração numa frequência de doze oscilações por minuto.

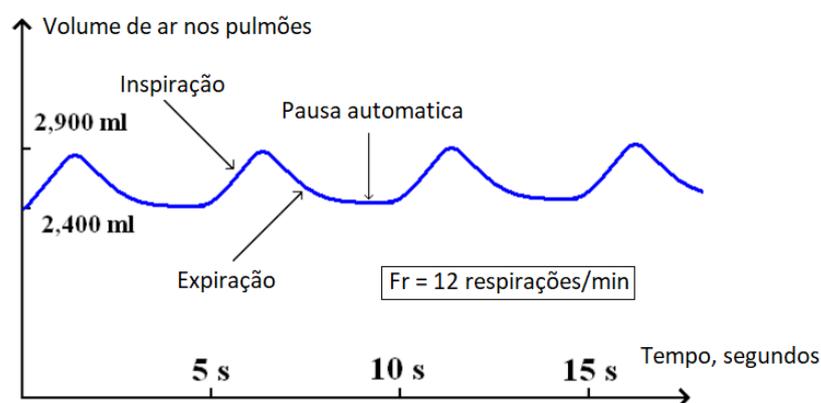


Figura 8 – Oscilações da respiração natural. Uma media de doze ciclos por minuto que alternam entre inspiração e expiração
Fonte: <http://respiracionnormal.org/>

Segundo o Instituto do Sono, cerca de 77% dos paulistanos sofrem algum distúrbio do sono, e 32,9% destes são portadores de apneia. A apneia é a parada respiratória devido a obstrução das vias aéreas num determinado período do sono. De acordo com a pneumologista Dra. Luciana Palombini, existem fatores genéticos e comportamentais que geram este evento como: a preocupação, o estresse e até mesmo o ambiente noturno (FIOCRUZ, 2019).

Para extrair este dado do usuário, será implementado o sensor de frequência respiratória para detectar o comportamento da respiração noturna e também possivelmente eventos de apneia.

2.3 MICROCONTROLADOR E MÓDULO *BLUETOOTH*

O ESP32, mostrado na Figura 10, é um SoC com microcontrolador integrado, *Wi-fi* e *Bluetooth*. Com grande aplicação para a indústria, dispositivos eletrônicos para vestir (ou *wearables*, no termo popular em inglês) e aplicações para Internet das Coisas (IOT - *Internet of Things*). Para melhorar as características de comunicação, ele é um dispositivo de baixo consumo de energia, graças à combinação de diversos tipos de *softwares* proprietários. O ESP 32 também é capaz de comunicar-se via *Bluetooth* ou *Wi-fi* com facilidade através das interfaces SPI/SDIO ou I2C/UART (ESPRESSIF, 2019). Além de todas estas características, o ESP 32, no momento do início do trabalho, tinha um preço acessível quando comparado aos concorrentes. Por essa e todas as demais características já citadas nesta seção, o kit de desenvolvimento ESP 32 DEV KIT V1 foi escolhido para ser aplicado neste trabalho.

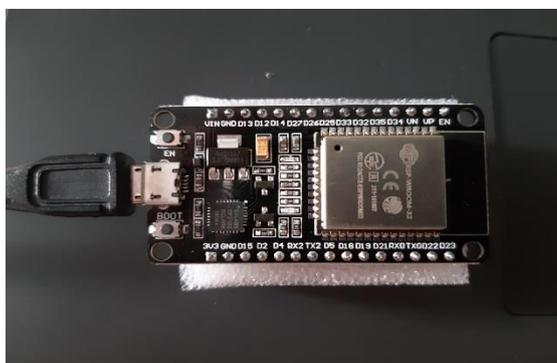


Figura 10 – Módulo de desenvolvimento ESP32 DEV KIT V1.
Fonte: Os Autores

2.4 *SMARTPHONE* E SISTEMA OPERACIONAL

Uma vez que os sensores captam as grandezas físicas do paciente em observação, estes comunicam-se com o microcontrolador responsável por condicionar os dados lidos. Em seguida, estes dados necessitam ser analisados e exibidos para o usuário. Diversas possibilidades foram consideradas para estas etapas, como usar microcontroladores com mais poder de processamento, displays

gráficos ou transmitir os dados para um computador com um *software* dedicado. Porém, *smartphones* e tablets mostraram-se muito eficientes neste aspecto por vários motivos, como facilidade de comunicação, preço, acessibilidade e poder de processamento. Isso tudo além do fato de que a maioria do público alvo deste projeto já tem um *smartphone*, dessa forma não haveria a necessidade de adquirir mais um equipamento dedicado.

Em 2016, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), cerca de 138 milhões de brasileiros tinham telefone móvel celular para uso pessoal e 78,9% destes tinham acesso à Internet. Além disso, da população que tinha acesso à internet, 94,2% a utilizava para enviar ou receber mensagens de texto, voz ou imagens por aplicativos diferentes de e-mail (IBGE, 2019). Com isso, pode-se concluir que uma expressiva parcela da população tem acesso a *smartphones*, fato que se traduz em um grande potencial de massificação para este projeto. Além disso, hoje é possível encontrar no mercado aparelho móveis com versões muito recentes do *Android* por preços acessíveis, como o Positivo Twist Mini S431, que possui sistema operacional *Android* 8.1, tecnologia *Bluetooth* 4.0 com suporte a *Bluetooth Low Energy* (BLE) e que pode ser encontrado no varejo por preços na faixa dos R\$ 249,90. Tendo isso em vista, o próximo passo foi definir qual seria a plataforma de desenvolvimento do aplicativo, mais especificamente, qual seria o sistema operacional no qual o aplicativo seria executado. Para definir este parâmetro, a popularidade foi o fator decisivo. A Tabela 2 mostra que, em 2017, 85,9% dos *smartphones* adquiridos vinham com sistema operacional *Android* (GARTNER, 2019).

Tabela 2 - Sistemas operacionais dos *smartphones* adquiridos no Brasil em 2017.
Fonte: GARTNER, 2019.

Sistema Operacional	Unidades (2017)	Quota de Mercado (2017)(%)	Unidades (2016)	Quota de Mercado (2016)(%)
<i>Android</i>	1.320.118,1	85,9	1.268.562,7	84,8
iOS	214.924,4	14,0	216.064,0	14,4

3 DESENVOLVIMENTO

É preciso validar cada etapa do projeto separadamente. Portanto, foram implementados todos os sensores de forma individual utilizando outros meios de processamento como microcontroladores Arduino e Raspberry PI 3 (não diretamente no ESP32 ou *Android*) para viabilizar uma validação construtiva mais dinâmica dos sensores. Idem à instrumentação, a transmissão de dados do ESP ao *Android* foi validada usando potenciômetros e comunicação com sensores digitais através de bibliotecas I2C. No âmbito da comunicação sem fio, do ESP ao *Android*, foram usados componentes para simular as naturezas lidas pelos sensores. Um exemplo disso foi o uso do sensor digital BMP180 e um potenciômetro para validar uma comunicação analógica entre os dispositivos.

3.1 ELIMINANDO NÍVEIS SIGNIFICATIVOS DE RUÍDO USANDO A MALHA DOS CABOS DE COMUNICAÇÃO.

Conforme foi sendo confeccionado o sensor de batimentos cardíacos, foram surgindo ruídos devido ao comprimento do fio que liga o sensor ao microcontrolador. Por ter um comprimento de dois metros e meio, ele fica suscetível a interações eletromagnéticas externas do sistema que podem interferir nos sinais elétricos quando são enviados ao microcontrolador. Uma análise inicial foi implementada para validar um modo de isolamento das ondas eletromagnéticas: com o algoritmo de leitura do sinal analógico e um potenciômetro. O potenciômetro foi usado como divisor de tensão e foram feitos dois testes de possíveis conexões com o cabo. No primeiro teste, foi utilizada uma conexão com a via simples e foram coletados valores do AD e armazenados numa planilha. Depois foi feito o mesmo procedimento com potenciômetro, mas usando a malha do cabo como GND. Conforme a Figura 11, os resultados foram bem diferentes entre ambos, pois a opção que usa a malha mostrou menos ruídos comprovando que há menos sensibilidade a falhas durante a leitura dos sinais dos sensores.

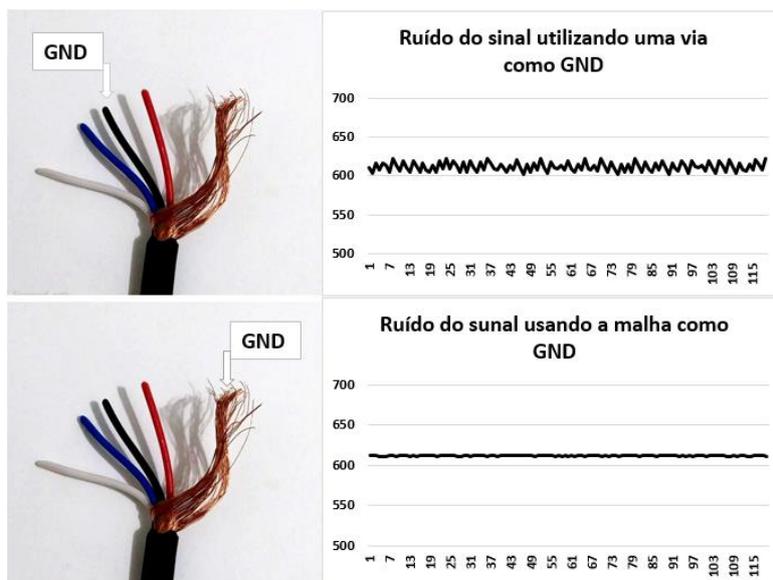


Figura 11 - Ensaio realizado para isolamento de ondas eletromagnéticas.
Fonte: Os autores

3.2 IMPLEMENTAÇÃO DO SENSOR DE BATIMENTO CARDÍACO

A importância do sensor de batimentos cardíaco para uma coleta noturna é verificar o grau de batimento conforme a noite passa. Porque quando houver um aumento no batimento pode se desconfiar que o corpo está passando por atividade de estresse. O batimento cardíaco médio é medido em BPM (Batimentos por minuto) e é diretamente proporcional a atividade física que o usuário está sujeito.

Para confeccionar um sensor, foi usado um sistema de LEDs com incidência luminosa constante e um fotodiodo. Seu funcionamento baseia-se na reflexão luminosa dos vasos sanguíneos em um fotorreceptor. Quanto há um batimento no coração, o fluxo sanguíneo aumenta ocorrendo uma dilatação momentânea nos vasos sanguíneos. Logo, com o aumento do fluxo, maior é a concentração de hemoglobina e maior luminosidade é absorvida pelo sangue, conseqüentemente menor luminosidade é refletida no fotodiodo. Sendo assim, é possível ter uma variação do feixe de luz no fotorreceptor proporcional as batidas do coração devido à variação periódica do fluxo sanguíneo.

Para fazer uma validação do sistema acima descrito foram realizados dois testes em bancada. O primeiro através da confecção de LEDs ultravioleta como emissores e um LDR (Light D Resistor) como receptor para detectar a luz refletida pelos vasos sanguíneos. Foram feitas seis diferentes concentrações de tintas em um

tubo de ensaio (tinta guache com água). A posição dos componentes ficou como na Figura 12, em uma angulação de 90 graus. O circuito de condicionamento do LDR resultou uma tensão diferente e proporcional a concentração de cada amostra do tubo de ensaio conforme mostrado na Figura 13.

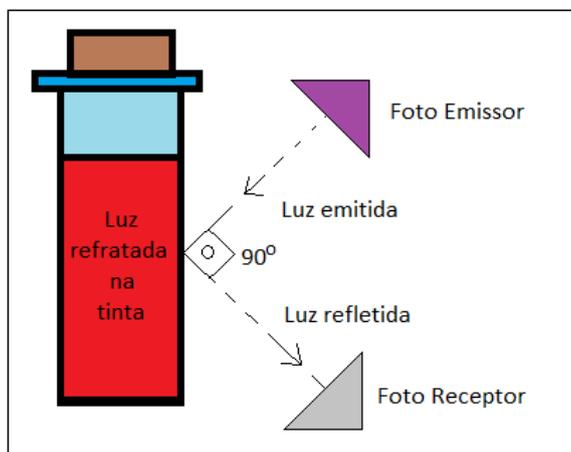


Figura 12 – Ilustração dos testes feitos em laboratório para validação do sistema óptico de reflexão da luz e sua proporcionalidade à concentrações de tinta.
Fonte: Os autores.

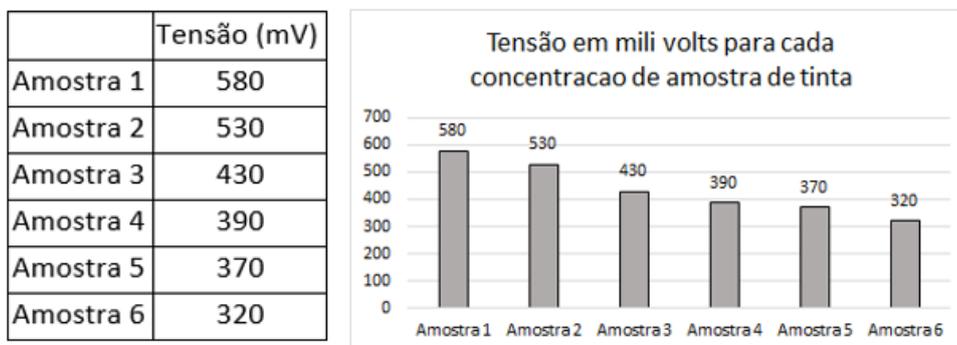


Figura 13 – Tabela e resultados dos sinais de tensão lidos para diferentes amostras de tinta.
Fonte: Os autores.

Em uma segunda validação foi utilizado um fotoemissor e um fotodiodo junto a um osciloscópio. Eles foram alinhados horizontalmente na ponta do dedo dos autores sendo que o fotodiodo fica diretamente sendo foco do fotoemissor. O resultado é mostrado na Figura 14, onde são mostradas as oscilações da variação luminosa no fotorreceptor - mesmo com um foto diodo e um fotorreceptor já foi possível detectar as oscilações luminosas proporcionais ao batimento cardíaco.

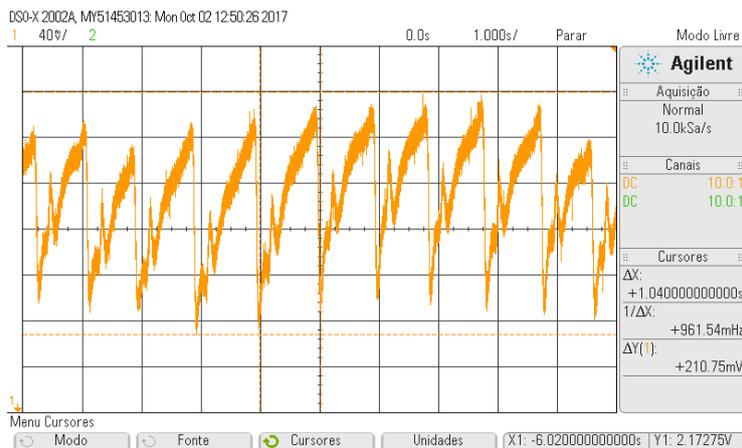


Figura 14 – Resultado da variação luminosa visto na tela do osciloscópio.

Fonte: Os autores.

Como a resposta do fotorreceptor foi possível de obter sem uso de um circuito de aquisição, somando com a praticidade do sensor de batimentos cardíacos comercial (Figura 15) devido a não necessidade de confeccionar uma placa de circuito impresso que elevando o custo com mão de obra, foram feitos testes e obtidos resultados satisfatórios. Com auxílio de um microcontrolador arduino, foram feitas coletas para quantificar os valores lidos pelo terminal analógico do sensor em alta velocidade (aproximadamente 200 amostras por segundo).

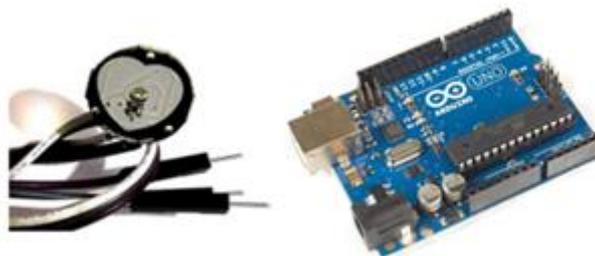


Figura 15 - Sensor óptico à esquerda usado para detecção de batimento cardíaco e microcontrolador arduino à direita usado para fazer coleta das variações luminosas recebidas pelo fotorreceptor do sensor óptico

Fonte: Os autores

Os valores analógicos lidos pelo Arduino foram colocados em uma planilha no excel para gerar um gráfico de análise (Figura 20) e mostram as variações oscilando de acordo com a batida do coração. O formato da onda foi descritivamente similar aos sinais de batimento cardíaco mostrados no livro de fisiologia médica GUYTON E HALL

(Figura 17) que descreve as ondas P, R e T assim como adquirido nos valores analógicos lidos pelo Arduino.

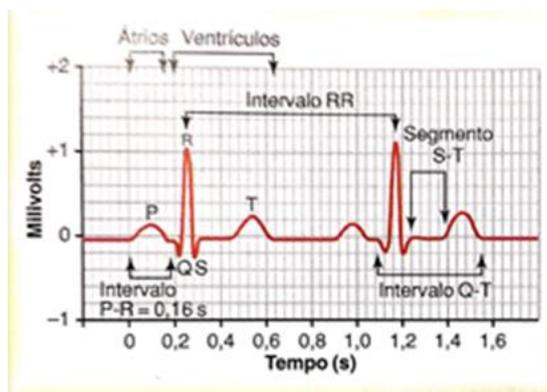


Figura 16 – Sinal fisiológico do batimento cardíaco de uma pessoa.

Fonte: GUYTON e HALL. Fisiologia médica, 2006.

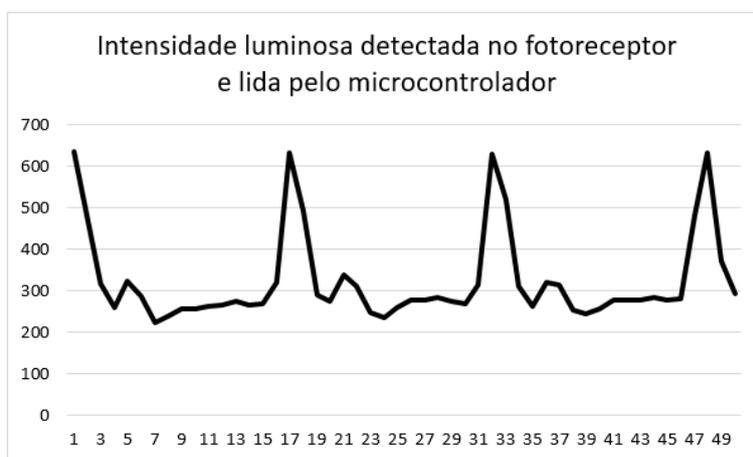


Figura 17 - Intensidade luminosa variando conforme os batimentos cardíacos.
Fonte: Os autores.

Para obter a frequência cardíaca, foi usado o seguinte procedimento. O algoritmo detecta qual o maior e menor dos valores armazenados no vetor (Sinais analógicos lidos pelo microcontrolador que variam conforme os batimentos cardíacos). Com esses valores são extraídos o valor médio momentâneo. Foi considerada uma janela de 3 segundos de coleta para esta análise. Nesses três segundos são feitas sessenta coletas com cinquenta milissegundos de intervalo entre cada uma,

resultando um vetor de sessenta elementos. A Tabela 3 abaixo ilustra a quantidade de elementos (neste caso sessenta) com os valores analógicos armazenados.

Tabela 3 – Valores analógicos proporcionais à intensidade luminosa lida como amostras armazenados em sessenta índices do vetor.

Fonte: Os Autores.

cont	0	1	2	3	4	...	57	58	59	60
AD	320	350	356	336	319	...	318	315	305	298

O algoritmo faz uma análise de todos os elementos para registrar qual o valor médio momentâneo entre o maior e menor elemento. Depois é feita uma verificação se os elementos iniciais estão acima ou abaixo do valor médio. Caso estejam acima, o algoritmo aguarda a onda descer para que na subida registre o valor da sequência do vetor (cont). Assim é feito com a segunda subida. O exemplo da Figura 18 mostra graficamente o que o algoritmo enxerga após fazer uma análise dos elementos do vetor. Nesse exemplo é ilustrado o sequenciamento de leitura analógica formando a forma do batimento cardíaco.

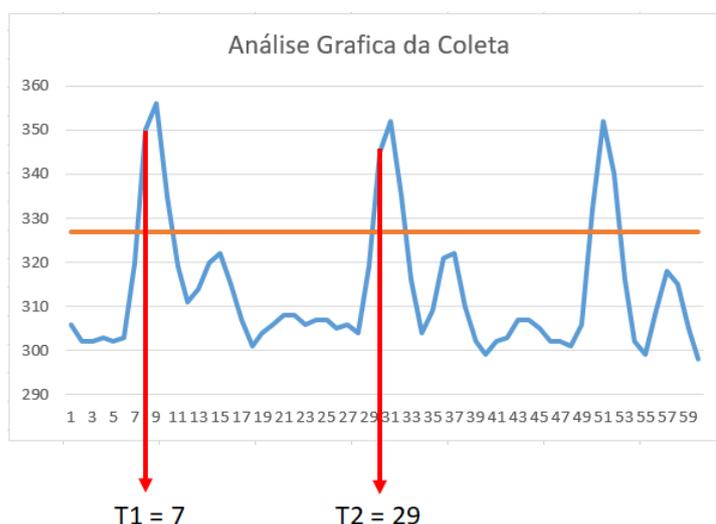


Figura 18 - O algoritmo armazena a sequência onde o elemento AD pela primeira vez fica maior que o valor médio (T1). Quando é registrado a segunda vez (T2), o algoritmo encerra seu cálculo fazendo a diferença entre as sequências T1 e T2.

Fonte: Os Autores.

O diagrama da Figura 19 mostra os procedimentos do algoritmo que resultaram o batimento cardíaco médio:

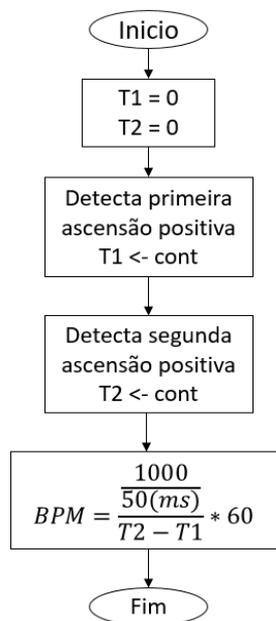


Figura 19 – Diagrama mostrando as etapas que o algoritmo calcula o batimento cardíaco médio
Fonte: Os Autores.

O algoritmo detecta a primeira ascensão do sinal e faz o registro do índice do vetor como T1. Após armazenar esse registro outra variável com a mesma função, agora T2, recebe o índice do vetor na segunda ascensão. A diferença entre os dois pontos, ou os dois índices do vetor, é usada para calcular o período do sinal. Como o programa implementado conhece o tempo entre as coletas durante o registro (período entre duas amostras de leitura do AD é igual a cinquenta milissegundos), o batimento cardíaco sai como resultado principal. Os resultados do sensor de batimento cardíaco foram satisfatórios. Não se mostrou viável para detectar sinais dos batimentos durante um teste de esteira por causa do ruído gerado pelo impacto das passadas, portanto é sensível a movimentos bruscos, mas tem uma validação aprovada, pois para medir os batimentos cardíacos durante o repouso não foi susceptível a ruídos tão frequentes como mostra a Figura 20. O sensor de batimentos foi colocado na ponta do dedo dos autores coletando dados do batimento cardíaco médio durante todo o sono.

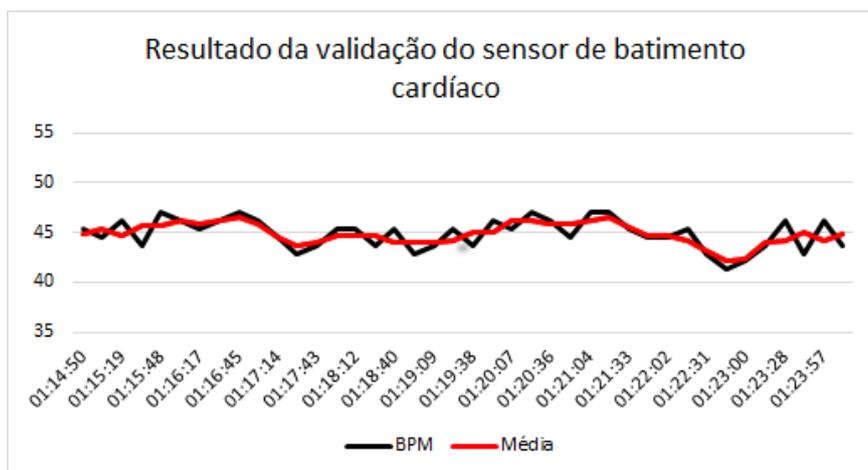


Figura 20 - O gráfico acima mostra uma aquisição de dados ao longo de nove minutos de uma coleta noturna.
Fonte: Os Autores.

3.3 IMPLEMENTAÇÃO E CONFECÇÃO DO SENSOR DE FREQUÊNCIA RESPIRATÓRIA

A importância do sensor de frequência respiratória (medido em IRM ou Índice de Respiração Média) é bem grande, pois um dos principais diagnósticos da qualidade do sono é a frequência respiratória. Ela pode ser normal mantendo uma frequência baixa que é proporcional ao grau de atividade corpórea, mas também pode haver irregularidades como a apneia. Para detectar casos de apneia é fundamental um sensor de frequência respiratória. Foi utilizado o sensor digital da Bosch BMP180, um sensor digital de pressão e temperatura. Através da comunicação I2C com um microcontrolador, o sensor pode enviar tempos em tempos valores alternados de temperatura e pressão.

O BMP180 foi inserido dentro de um plástico de ar e fixado em uma cinta confeccionada com passador e velcro como mostra na Figura 21. Conforme o usuário inspira e expira, sua caixa torácica expande comprimindo o plástico de ar com o sensor dentro. Essa variação na compressão é facilmente detectada pelo sensor em níveis de pressão pelo tempo.



Figura 21 – Cinta torácica onde foi anexada o invólucro de ar que contém o sensor de pressão BMP180.
Fonte: Os autores.

Utilizando o Raspberry para validação do protótipo, foi implementado um algoritmo que a cada segundo recebe quatro amostras de pressão do sensor BMP, tempo suficiente para avaliar a frequência respiratória neste teste. Essas coletas são enviadas para um arquivo de texto. Foi implementado um disparo usando uma alavanca mecânica para dar início e final à coleta. A Figura 22 mostra a chave mecânica ao lado do fluxograma de coleta e armazenamento de dados em um arquivo de texto.

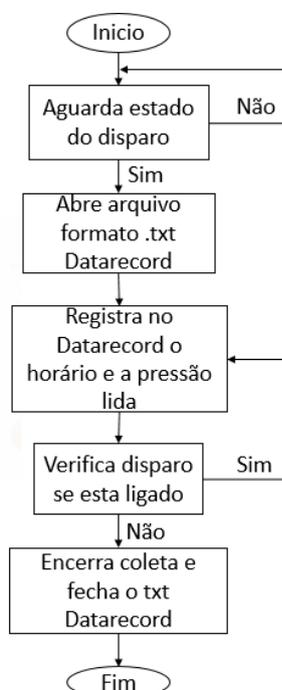
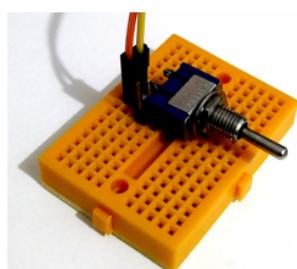


Figura 22 - Fluxograma do algoritmo disparado por uma chave mecânica para validar uma coleta noturna.
Fonte: Os Autores.

O algoritmo implementado aguarda a resposta da chave mecânica para iniciar e finalizar novamente a coleta. Numa experiência noturna, a cada quatro segundos, o algoritmo faz uma leitura do sensor de pressão que estava no invólucro de ar. A Figura 23 mostra a variação da pressão lida pelo sensor digital ao decorrer do tempo. Percebe-se que a variação tem a mesma característica e perfil da frequência respiratória.

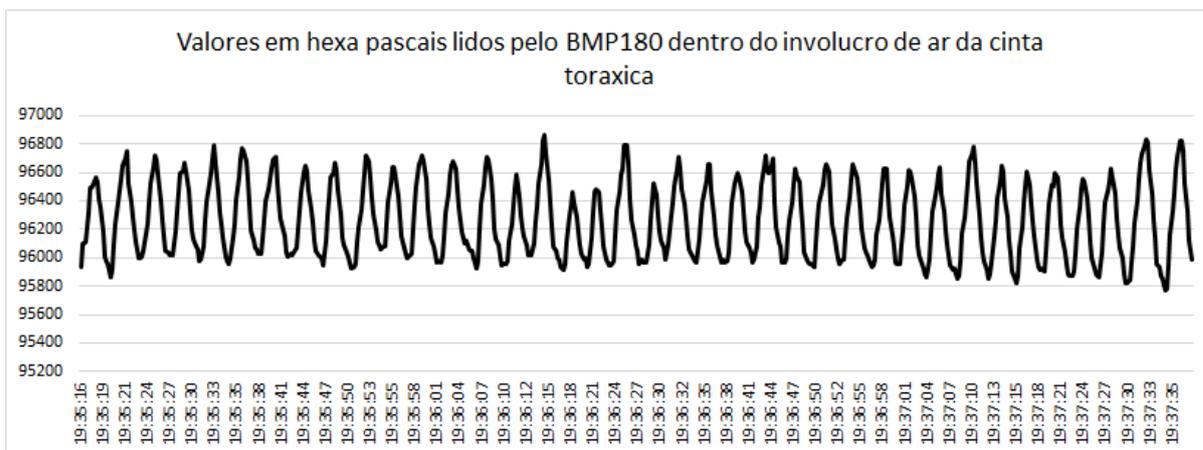


Figura 23 – Resultado da variação da pressão interna do invólucro de ar da cinta torácica. Fonte: Os autores.

A Figura 23 mostra a variação da pressão da caixa torácica que reflete no invólucro de ar que é lida pelo sensor e interpretada pelo algoritmo. Como pode ser observado visualmente no gráfico, a variação da pressão lida pelo sensor BMP varia de acordo com a frequência respiratória. O seguinte passo foi implementar um algoritmo para o microcontrolador calcular a frequência respiratória direta. Para isso, foram implementadas duas ideias distintas. A primeira baseia-se na inversão do sinal da derivada no momento das cristas e vales da onda periódica (como observado no sinal da frequência respiratória). Sempre que houver uma inversão, uma variável é acionada como indicador. A frequência que essa variável altera é duas vezes a frequência respiratória (porque uma mudança ocorre no vale e outra na crista - duas mudanças em um mesmo período de onda). O resultado não foi viável, pois como pode ser observado nas oscilações da Figura 23, existem picos duplos em algumas cristas como é mostrado melhor na Figura 24.



Figura 24 - incidente recorrente que causa inversões duplas na derivada da reta tangente na mesma crista do sinal.
Fonte: Os Autores.

Nos vales do sinal da frequência respiratória onde o volume de ar é grande, foi observado um ruído curioso que se repetia com frequência ao decorrer da coleta. No momento da expansão, o sensor fica mais sensível a ruídos sendo possível captar também a vibração da batida do coração.

A ideia definitiva foi agregar o mesmo algoritmo usado para detectar os batimentos cardíacos médios, pois o mesmo comportamento oscilatório é observado em ambos sinais vitais. Apesar de serem semelhantes, a frequência é diferente porque o batimento cardíaco acontece mais rápido - e três segundos são suficientes para ter certeza que na amostra coletada há um ciclo completo. Já na frequência respiratória, o período das ondas se repetem num intervalo de tempo maior, portanto exige uma dedicação maior para coletar um ciclo respiratório completo. Estimando uma respiração média de doze a dezesseis ciclos por minuto, quatro segundos são suficientes para detectar um período de respiração completo. Entretanto, levando em consideração um sono profundo e possibilidade de apnéia, dez segundos foram atribuídos para fazer a coleta de uma amostra. Para uma validação mais concreta, foi feita uma coleta de vinte minutos sobre a frequência respiratória dos autores. A Figura 25 mostra as variações da frequência respiratória com a sua média.

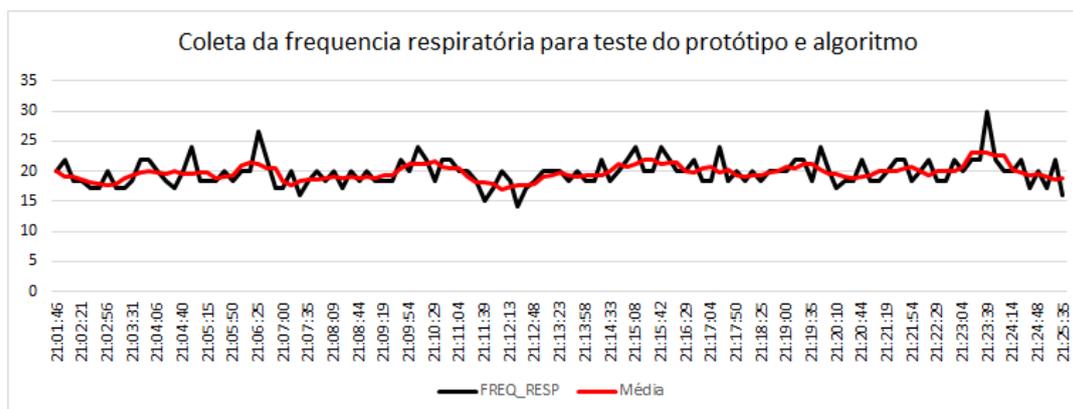


Figura 25 - Resultado de uma coleta curta para validar o sensor de frequência respiratória. A interferência gerada no sinal pelo ruído vem de movimentos físicos enquanto a coleta estava sendo feita.
Fonte: Os Autores.

No resultado coletado, mostrado na Figura 29, apesar da média compreender uma faixa de quinze a vinte e cinco IRM (Índice de Respiração média - ciclos por minuto), são valores condizentes com as referenciadas no livro de fisiologia médica (GUYTON e HALL, 2006). Durante outro teste (Tabela 4), foram contados manualmente quantos ciclos de respiração que o autor realizou, sendo um valor bem próximo ao medido.

Tabela 4 - Resultado das frequências respiratórias lidas do autor e sua média no minuto completo. Durante o intervalo de teste, manualmente, foram contabilizados dezesseis ciclos completos bem como a média mostrada na tabela.

Horario	IRM
21:34:10	16
21:34:22	16
21:34:33	14
21:34:45	15
21:34:57	17
Média	16

3.4 IMPLEMENTANDO O SENSOR DE BRUXISMO

O bruxismo é uma desordem originada de diversos fatores, entre eles ansiedade e mau gerenciamento do estresse diário. Para isso foi utilizado o mesmo sensor de pressão digital para verificar se houve alguma pressão exercida pela mandíbula do usuário. Existem vários níveis de bruxismo, mas por enquanto a

importância do sensor é verificar se há algum sinal que possa ser correlacionado com este problema. Seja ele alteração na frequência respiratória, batimento cardíaco médio ou até mesmo movimentação involuntária noturna.

Foram adquiridas placas odontológicas de silicone que se moldam à arcada dentária após imersos em água quente. Duas dessas placas foram usadas para fazer um molde especial que se sobrepõe formando um espaço vazio entre ambos, como mostra na Figura 26. Neste espaço vazio foi inserido o sensor digital de pressão. Apesar de ser intrusivo, foi uma maneira rápida e eficiente de ser implementado um sensor de bruxismo.

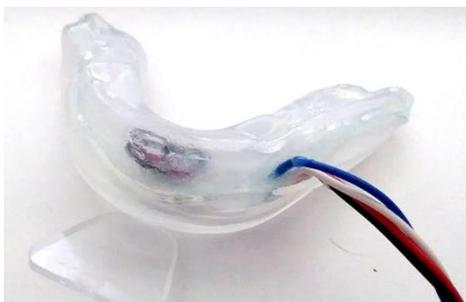


Figura 26 - Imagem do sensor de bruxismo. Formado por duas camadas de silicone moldadas com a arcada dentária do usuário e que englobam o sensor de pressão digital. Nela há quatro fios de funcionamento e comunicação dos dados.

Fonte: Os Autores.

Foi realizada a coleta frequente dos valores de pressão do sensor digital. Essa pressão lida pelo BMP180 é gerada pela mordida do usuário e pode ser verificado na Figura 27 uma instabilidade próxima às 22:20 que revela a ocorrência de uma mordida.

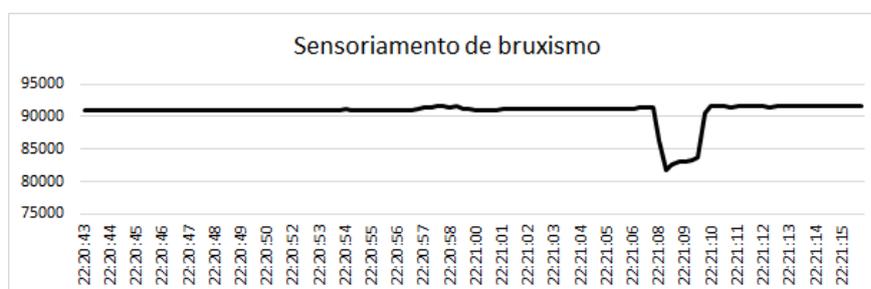


Figura 27 – Simulação da ocorrência de bruxismo captada pelo sensor desenvolvido.
Fonte: Os autores.

3.5 IMPLEMENTANDO O SENSOR DE MOVIMENTAÇÃO NOTURNA

A detecção do movimento é uma variável bem importante, pois quanto mais o usuário se mexe durante a noite, menores são as chances de ter uma noite tranquila de sono. Para extrair este dado foi usado o acelerômetro digital MPU 8050. O chip possui um acelerômetro, transdutor de temperatura e um giroscópio também, e suas leituras alternam conforme requisitado pelo protocolo de comunicação I2C.

Inicialmente foi confeccionado uma cinta para alocar o sensor, a qual é parecida com a do sensor de frequência respiratória, pois busca-se diminuir o incômodo sentido pelo usuário desta forma. Para validar um algoritmo foi aplicado no Raspberry um código em python que capta os valores dos respectivos eixos do acelerômetro e compara com os valores lidos três segundos antes, conforme exemplificado também na Figura 28. Caso o algoritmo note uma alteração significativa nesses eixos, significa que naquele período de tempo houve uma movimentação, por consequência o contador de movimentos é incrementado.

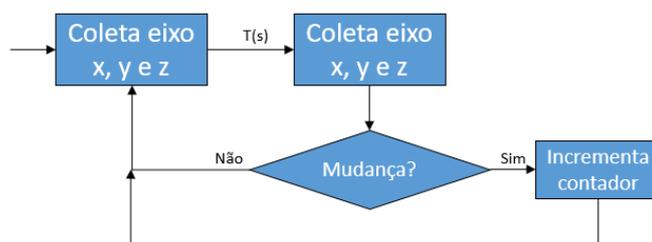


Figura 28 – Funcionamento do algoritmo para detecção de movimentos noturnos.
Fonte: Os autores.

Para validação, foi realizado uma coleta durante uma noite de sono do autor, foi possível contabilizar quantas vezes houve movimentação durante o intervalo da coleta e também qual foi o horário que ocorreram esses movimentos. Na Figura 29, durante essa coleta noturna, foram registrados os horários e as intensidades dos movimentos captados pelo acelerometro.

Resultado da Coleta				
Hora	Situação	Intensidade	movimento(acumulado)	temperatura
		(...)		
01:16:40	Movimento Detectado	1,5	22	34,8
01:17:11	Movimento Detectado	1,0	23	34,8
03:13:37	Movimento Detectado	2,3	24	37,3
03:13:42	Movimento Detectado	4,0	25	37,3
03:13:47	Movimento Detectado	9,3	26	37,3
03:41:31	Movimento Detectado	3,3	27	37,1
03:41:36	Movimento Detectado	9,1	28	37,0
		(...)		
04:52:44	Movimento Detectado	9,2	34	37,7
04:52:49	Movimento Detectado	2,7	35	37,6
05:38:48	Movimento Detectado	1,1	36	38,1
05:38:53	Movimento Detectado	1,5	37	38,1
05:38:58	Movimento Detectado	13,4	38	38,7
		(...)		

Figura 29 – Resultado de uma coleta noturna realizada nos autores.
Fonte: Os autores.

Como mostrado na Figura 34, entre o horário de 01:17 até 03:13 não foi detectado movimento significativo durante o sono do usuário, logo conclui-se que houve um período estável de sono durante a noite. Durante toda coleta, os intervalos que forem detectados como sono profundo (sem movimentação noturna) serão somados e contabilizados como total de tempo estável de sono. Além do horário e a intensidade do movimento, são como complementos: a variável do contador de movimentos e temperatura coletada pelo acelerômetro. Logo, conclui-se com a validação acima que o sensor de movimentação noturna se torna relevante quando correlacionado com outros sinais. Pois enquanto o usuário está passando por um sono profundo, outros sensores fazem um papel fundamental na coleta dos outros sinais.

3.6 CALIBRAÇÃO DAS INTENSIDADES DE MOVIMENTAÇÃO NOTURNA

A intensidade do movimento depende muito da variação angular aplicada no acelerômetro. Utilizando um transferidor, foram feitas coletas dos valores enviado pela comunicação I2C em função do ângulo no qual o componente estava em relação ao plano horizontal. A curva de caracterização teve um perfil logarítmico como pode-se observar na Figura 30, mas pode ser aproximada em uma reta devido a faixa de alcance partindo do zero a noventa graus. Outro detalhe que facilita usar essa aproximação é o uso dos outros dois eixos que atuam como complemento para detectar os três graus de liberdade referente à movimentação.



Figura 30 - Curva de resposta das intensidades atribuídas pelo sensor MPU6050 em resposta a um ângulo isolado.
Fonte: Os Autores.

Logo, foram consideradas como movimentações os valores maiores que 1,0 (em unidade de resposta do sensor). Pois se entre duas verificações houver uma diferença maior ou igual a 1,0, significa no mínimo dez graus de movimentação. Portanto, se durante a noite o sensor de movimentos detectar um estímulo acima de dez graus, o algoritmo atribui uma movimentação registrada.

3.7 APLICAÇÃO DO SENSOR DE TEMPERATURA PARA VALIDAÇÃO DOS SINAIS VITAIS

Como os sensores BMP180 (pressão digital) e MPU 6050 (Acelerometro e giroscópio) possuem transdutores de temperatura embarcados, foram usados então como indicadores de estabilidade para verificar o grau de contato com o corpo humano. Ou seja, a faixa do sensor de frequência respiratória, por exemplo, fica constantemente em contato com o peito do usuário. Qualquer mal posicionamento ou movimentação da faixa em relação ao peito pode ser identificada devido a variação da temperatura coletada naquele período da noite. Assim como o usuário, a cinta do sensor de movimentos e da frequência respiratória estão sujeitas a movimentação física. Essa movimentação pode alterar o posicionamento ou contato dos sensores com o corpo humano e conseqüentemente gerar um grau de inviabilidade da coleta dos sinais vitais. O exemplo abaixo (Figura 31) foi visto durante uma coleta dos dados da frequência respiratória. Entre as duas e as quatro e meia da manhã, houve uma desestabilização da temperatura lida pelo sensor. Isso atribui uma incerteza nos valores de frequência respiratória naquele mesmo intervalo tempo, portanto demandando mais atenção quanto a veracidade dos dados.

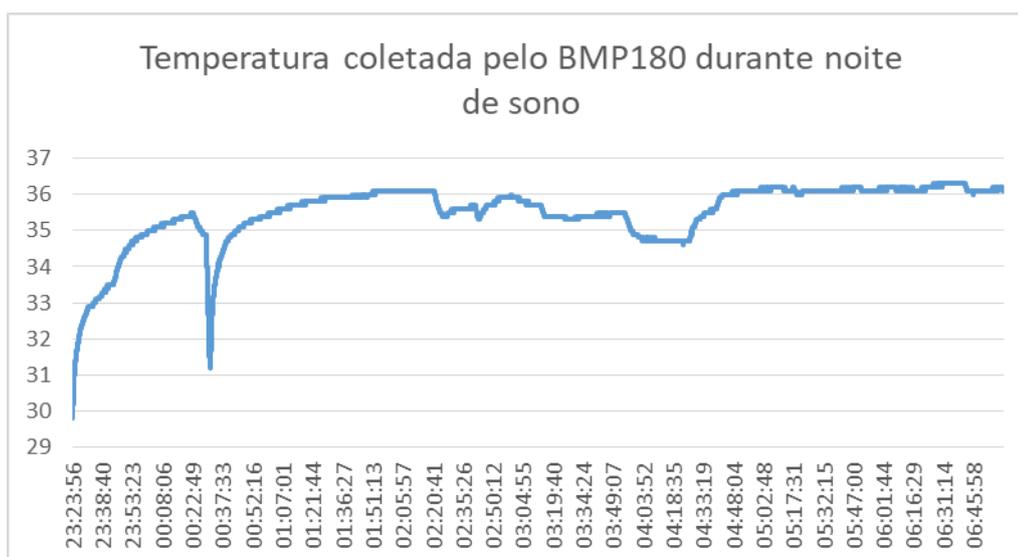


Figura 31 – Variação da temperatura coletada no interior do invólucro de ar do sensor de frequência respiratória. Houve um desacoplamento próximo à meia noite e meia e no período de duas e meia da manhã até as cinco horas.
Fonte: Os autores.

3.8 APLICATIVO *ANDROID*

O aplicativo é o programa executado no sistema operacional *Android*, ele será responsável por todas as tarefas feitas no *smartphone*. Os diversos arquivos que ele contém, sejam editados pelo desenvolvedor ou gerados pelo ambiente de desenvolvimento utilizado, são compactados no formato APK. Em seguida, o aplicativo é instalado no aparelho e estará pronto para ser executado.

Neste capítulo, os métodos, ferramentas e o desenvolvimento do aplicativo serão expostos.

3.8.1 Editor e outras ferramentas

Para a elaboração e edição do código do aplicativo, foi utilizado o programa *Android Studio*. Este é desenvolvido pela Google, empresa que também é responsável pela criação do sistema operacional *Android*.

O programa em questão é conhecido e considerado pela maioria dos desenvolvedores *Android* como a melhor plataforma para se trabalhar com este propósito. Ele possui diversas funcionalidades que aumentam a eficiência da escrita dos códigos. Além disso, possui recursos como o simulador de *smartphone*, onde é possível criar um *smartphone* virtual para executar e testar o aplicativo que está em desenvolvimento.

Outro programa utilizado foi o Microsoft Visual Code em conjunto com a ferramenta de repositórios GIT. Juntos, além de criar um repositório e trazer mais segurança e praticidade para o desenvolvimento, é possível visualizar as mudanças feitas em cada *commit* de forma muito intuitiva. Para aumentar ainda mais a segurança, foi criado um repositório online na plataforma GitHub.

3.8.2 Comunicação com o *Smartphone*

Após ser definida a plataforma de desenvolvimento do aplicativo, necessita-se definir a forma de comunicação (preferencialmente local, sem intermediadores) entre o microcontrolador que condiciona os dados dos sensores e o *smartphone* que executa o aplicativo. As possibilidades são:

- Wi-Fi
- *Bluetooth*
- Interface USB

A interface USB foi imediatamente descartada pois esforços deveriam ser tomados para que, ao decorrer do trabalho, houvesse mais facilidade e conforto para o usuário em relação à uma polissonografia convencional, desta forma, a ausência de fios entre os sensores e a central de análise foi fundamental.

Havendo restado apenas as comunicações *bluetooth* e Wi-Fi, ao invés de definir o tipo de comunicação, resolveu-se definir o microcontrolador, no caso, um kit de desenvolvimento ESP 32 (mais detalhes na seção 5.6). Isso devido ao fato de que o *System on a Chip* (SoC) utilizado possui os dois tipos de comunicação citados, permitindo a escolha deste em meio ao desenvolvimento, dependendo da conveniência. De qualquer forma, a conexão via *bluetooth* BLE é mais adequada, a princípio, por questões de velocidade no estabelecimento da conexão e consumo de energia.

3.8.3 Protocolo de comunicação

O protocolo utilizado na comunicação via BLE entre o *smartphone* e o ESP32 é o *Generic Attributes Profile* (GATT, ou Perfil de Atributos Genéricos), que por sua vez é baseado no *Attributes Protocol* (ATT, ou Protocolo de Atributos).

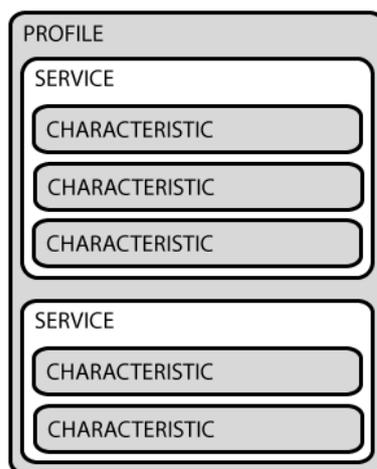


Figura 32 - Estrutura do perfil do GATT.
Fonte: <http://adafruit.com/>

Como pode-se observar na Figura 32, o protocolo em questão é definido em função do perfil das partes, porém este pode ser trocado a qualquer momento da comunicação. Isso porque os dispositivos podem ser hora cliente, hora servidor.

Depois de definido o perfil, o servidor cria um Serviço ao qual é atribuído um Identificador Único Universal (UUID, de Unique Universal Identifier). Este elemento é criado para organizar os valores atribuídos a ele no formato de uma ou mais *Características*. Estas também são identificadas por um UUID e podem conter apenas um valor, embora este possa ser um array e a *Característica* também possa conter Descritores. Estes últimos não foram utilizados neste trabalho.

Desta forma, foram implementados métodos nos dois dispositivos deste trabalho, o ESP32 e o *smartphone*, com o objetivo de estabelecer a comunicação entre eles. O fluxograma da Figura 33 abaixo descreve essas iterações.

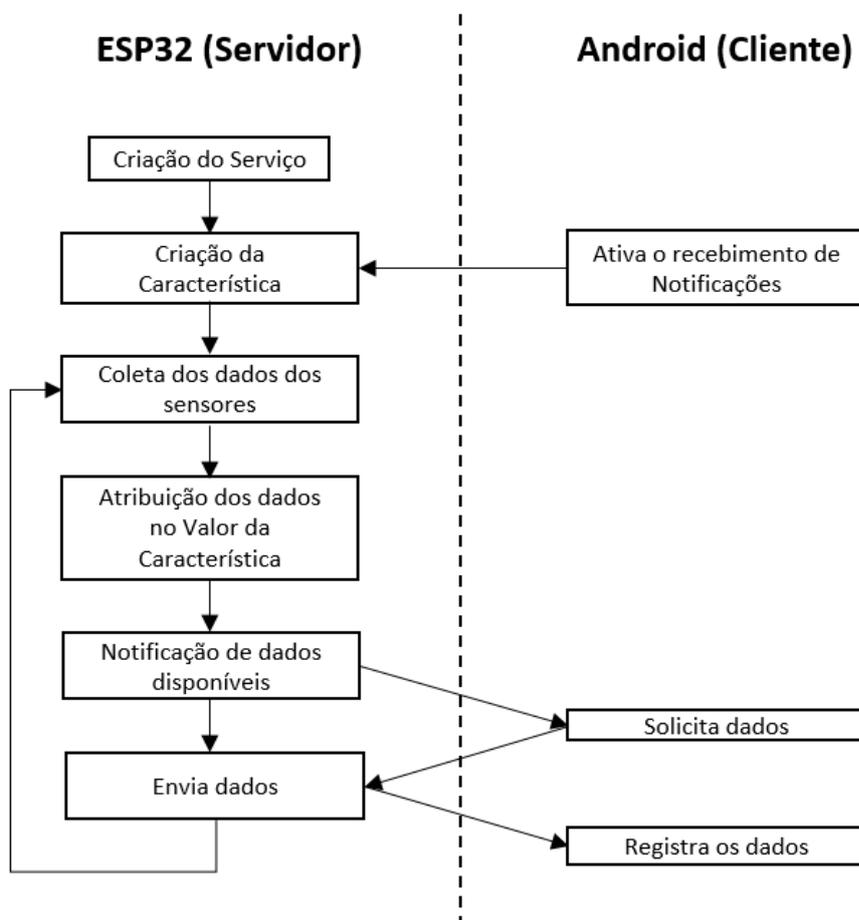


Figura 33 - Algoritmo para comunicação entre o ESP32 e o Android.
Fonte: Os Autores.

Uma vez que a característica está criada e a conexão entre o servidor e o cliente foi estabelecida, o algoritmo indica que os dados dos sensores serão lidos e atribuídos à característica e, em seguida, o servidor deve notificar o cliente de que há dados disponíveis. Porém, para que o *Android* receba esta informação, ele primeiro deve declarar que está interessado em receber tais notificações. Para isto é utilizado o único Descritor do sistema, o que ativa as notificações do lado do cliente.

3.9 ESTRUTURA DO CÓDIGO

Neste subcapítulo, será abordada a forma na qual foi escrito o código do aplicativo, assim como os recursos utilizados.

3.9.1 Módulo

Módulo é o conjunto de arquivos e configurações de compilação. Ele pode representar a estrutura principal do aplicativo assim como uma biblioteca, sendo o intuito atribuir mais organização para o projeto.

Dentro do módulo, que consiste em uma pasta, existem outros diretórios nos quais são armazenados arquivos específicos, na Figura 34 é possível observar estas pastas e quais arquivos são armazenados.

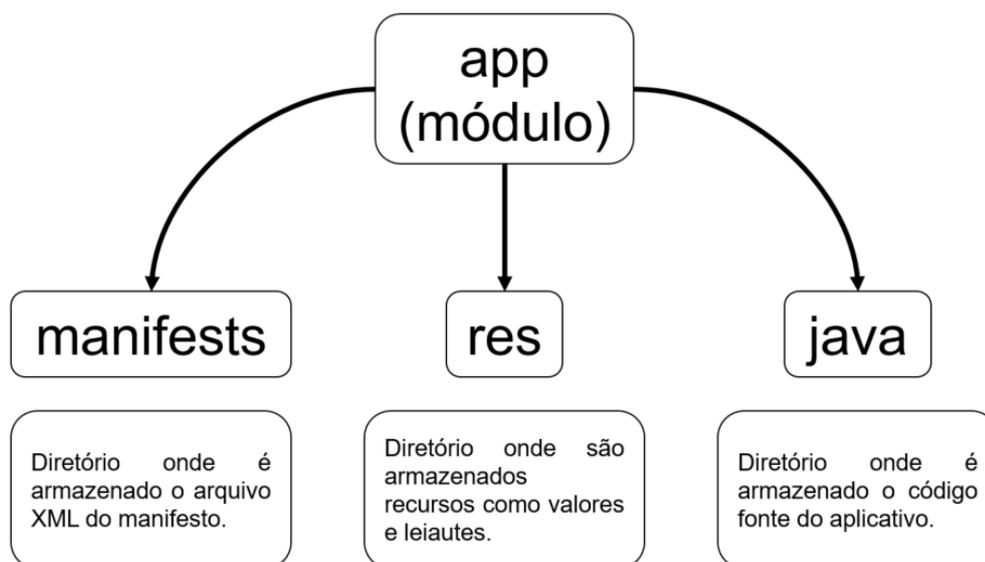


Figura 34 - Estrutura do módulo do aplicativo.
Fonte: Os Autores.

3.9.2 Manifesto

O manifesto do aplicativo é um documento XML (nomeado necessariamente de *AndroidManifest.xml*) que contém informações imprescindíveis para a execução do programa.

Desta forma, neste trabalho, foi necessário introduzir no manifesto as informações dos subcapítulos a seguir.

3.9.2.1 Permissões

As permissões necessárias são incluídas no manifesto para que, à medida em que o aplicativo é executado, o usuário seja consultado sobre elas. Caso as permissões requeridas sejam concedidas, o aplicativo terá acesso aos recursos.

Para pedir uma permissão no manifesto, basta incluir a seguinte linha de comando:

```
<uses-permission Android:name="Android.permission.NOME_PERMISSAO" />
```

onde *NOME_PERMISSAO* é o valor da constante para uma determinada permissão.

Foram utilizadas as seguintes:

BLUETOOTH: Permite a conexão com dispositivos pareados via *bluetooth*.

BLUETOOTH_ADMIN: Permite descobrir e parear dispositivos *bluetooth*.

ACCESS_COARSE_LOCATION: Permite o acesso à localização aproximada do dispositivo.

WRITE_EXTERNAL_STORAGE: Permite a escrita no armazenamento externo (no caso, não somente na região do armazenamento destinada exclusivamente para o aplicativo em questão).

3.9.2.2 Recursos de *hardware* e *software* necessários para aplicativo

Nesta parte do código, definem-se os elementos do *<uses-feature>*. O intuito deste trecho é de informar ao *Google Play* (loja de aplicativos da Google) quais recursos de *hardware* e *software* são utilizados nesta aplicação. Desta forma, o

aplicativo pode não ser listado em pesquisas feitas por dispositivos cujo *hardware* não contem alguma das funcionalidades necessárias. Um exemplo para este trabalho é a localização, requerida em função do BLE devido a implementação do rastreamento da localização de atividades físicas, como é o caso dos relógios da Garmin, já citados na introdução. Desta forma, caso este aplicativo seja adicionado ao *Google Play*, ele não será visível para dispositivos que não possuem GPS, por exemplo.

Neste trabalho, foi definido apenas o recurso de localização pelo GPS e armazenamento externo.

3.9.2.3 Atividades

A atividade (ou no termo original em inglês, *activity*) é algo singular no qual o usuário pode se focar. Ela se responsabiliza por criar uma janela na qual é construída a interface do usuário. As atividades também devem ser definidas no manifesto.

Neste trabalho, utilizam-se duas atividades. A primeira é responsável pelo descobrimento e conexão com o ESP32 via *bluetooth*. A segunda é quem faz o gerenciamento da coleta de dados dos sensores e exibe os resultados para o usuário.

3.9.2.4 Serviços

Serviços são os elementos responsáveis pela execução de operações de longo período enquanto não interagem com o usuário ou que suprem as necessidades de operação de outras atividades. Também são necessários no manifesto.

Neste caso, a classe *BluetoothLeService* é utilizada, portanto foi definida no arquivo em questão.

3.9.3 Recursos

A pasta *res* (do inglês *resources*) possui arquivos XML que contém valores, menus e *layouts* do aplicativo.

Nos arquivos referentes aos *layouts* e menus, foram definidas as características e posicionamento dos botões, caixas de texto, caixas de rolagem,

barras de progresso, gráficos, etc. Também foram definidas *strings* e outros valores, prática importante para manter o código organizado, tendo em vista que, quando os valores são chamados, isto é feito através do *ID*, uma espécie de apelido. Assim, caso seja necessário modificar algum valor, basta modificá-lo no arquivo XML, sem que seja necessário alterar o código fonte.

3.10 CÓDIGO DA APLICAÇÃO *ANDROID* EM JAVA

3.10.1 Atividade de busca de dispositivos próximos

Esta atividade é responsável pela tela inicial do aplicativo onde são listados os dispositivos *bluetooth* próximos. Ela foi criada utilizando-se a *ListActivity*, uma Interface de Programação de Aplicação (API) do *Android* que consiste basicamente em uma atividade em formato de lista.

Nesta classe Java, a primeira implementação mais importante é o *BroadcastReceiver*, devido ao fato de estabelecer a relação de dispositivos *bluetooth* próximos. Este tipo de método é muito comum na plataforma *Android* e é responsável por receber mensagens transmitidas pelo aparelho ou pelas aplicações (*broadcast messages*). Estas mensagens são muito úteis para a comunicação entre os elementos envolvidos. Neste caso, a implementação do *BroadcastReceiver* é feita para receber especificamente a transmissão do serviço *BluetoothLeService* (este serviço é descrito no capítulo 3.10.3) cuja mensagem indica que um novo dispositivo foi encontrado nas proximidades.

Depois da descoberta de um novo dispositivo, demais métodos foram implementados para incluir as informações deste na interface gráfica no formato de lista, como nome do aparelho, endereço *bluetooth* e do tipo de conexão utilizada, BLE ou *Bluetooth Classic*, sendo esta última o protocolo clássico *bluetooth*, onde a redução do consumo de energia não é a maior prioridade.

Tendo exibido os dispositivos próximo, a segunda implementação mais importante desta atividade é acionada, pois é ela que dá seguimento para a próxima aplicação estabelecer a conexão *bluetooth* com o aparelho. Trata-se do método *onListItemClick*. Ele é acionado através da interface gráfica quando o usuário seleciona um item da lista de aparelhos. Quando isto ocorre, o aplicativo salva as

informações do item selecionado, encerra o escaneamento *bluetooth* e inicia a outra atividade através do método *startActivity*, sendo este último passo o mais importante, tendo em vista que é ele que, de fato, inicia a seguinte atividade. A partir desta parte, a classe *DeviceControlBLEActivity* fica responsável pelos demais passos.

3.10.2 Atividade de controle de dispositivo conectado

DeviceControlBLEActivity é o nome dado à outra atividade do aplicativo deste projeto. A principal função dela é de exibir as mensagens da comunicação *bluetooth* com o ESP32. Isto é feito primeiramente utilizando a mesma API descrito na atividade anterior, o *BroadcastReceiver*. Neste caso, a atividade em questão recebe as mensagens de *broadcast* do serviço *BluetoothLeService*, porém as mensagens são diferentes. Os assuntos destas mensagens são os seguintes:

- Dispositivo conectado (*ACTION_GATT_CONNECTED*);
- Dispositivo desconectado (*ACTION_GATT_DISCONNECTED*);
- Serviço descoberto (*ACTION_GATT_SERVICE_DISCOVERED*);
- Dados disponíveis (*ACTION_DATA_AVAILABLE*).

De acordo com a mensagem de *broadcast* recebida, uma função será atribuída. Por exemplo, o botão de “conectar” é substituído pelo “desconectar” quando o dispositivo está conectado, ou seja, quando o *BroadcastReceiver* recebe a informação “*ACTION_GATT_CONNECTED*”. No momento em que há conexão estabelecida, ao mesmo tempo, o aplicativo faz uma escrita em um Descritor na *Característica* do ESP32, informando que aceita o recebimento de notificações.

Quando o usuário clica no botão “Iniciar” na interface da atividade, o aplicativo escreve uma mensagem na *Característica* do informando que as leituras devem começar. Ao mesmo tempo, o texto exibido no botão é substituído por “Parar”, e a cada mensagem que o aparelho *Android* recebe do ESP 32, o *BroadcastReceiver* é acionado e o resultado é impresso na tela. Também, os valores são escritos em um arquivo TXT, que é gravado na raiz do armazenamento interno do *smartphone*, utilizando-se o método *writeToFile*.

Ao clicar em “Parar”, é escrita uma mensagem na característica do ESP32 informando que o processo de leitura dos dados deve ser interrompido, mostrando

uma notificação na tela de que o registro foi encerrado. Além disso, inicia-se o processo de análise dos dados coletados, onde são encontrados os pontos máximos, mínimos e a média de cada série de dados. Também são adicionados na interface gráfica os gráficos e tabelas referentes aos valores calculados.

3.10.3 Serviço de controle da comunicação *bluetooth*

O serviço *BluetoothLeService.java* é uma implementação utilizada para facilitar a comunicação *bluetooth* com o dispositivo conectado. Esta classe possui uma diversidade de métodos que facilitam a comunicação do aplicativo com o ESP32.

A primeira implementação é a da *BluetoothGattCallback*, uma classe abstrata para a implementação de *callbacks* da API *BluetoothGatt*. Quatro métodos desta classe foram implementados.

O primeiro método é o *onServicesDiscovered*. Ele é responsável por detectar se há algum serviço do protocolo *GATT* criado no dispositivo conectado, como descrito no 3.8.3. Depois disso, ele envia uma mensagem de *broadcast* informando que o serviço foi descoberto (*ACTION_GATT_SERVICES_DISCOVERED*).

Seguindo a ordem do código, os próximos elementos são os *callbacks* de leitura e escrita em uma característica. O primeiro deles é o *onCharacteristicRead*, cuja implementação é muito simples, pois apenas invoca o método *broadcastUpdate*, responsável por enviar, neste caso, a mensagem de *broadcast* com o resultado da operação, se foi bem-sucedida ou não. O segundo *callback* é o que implementa uma função do protocolo utilizado, onde toda operação de escrita feita pelo cliente é seguida de uma operação de leitura. O método é o *onWriteCharacteristic*, que basicamente invoca a função *readCustomCharacteristic*, responsável pela leitura do valor da característica. Também há um método chamado *onCharacteristicChanged*, que também emite uma mensagem de *broadcast* informando a mudança da característica que chega por notificação do protocolo.

3.11 VERIFICAÇÃO DO CONSUMO DO SISTEMA E BATERIA

Para viabilizar que uma bateria se adeque com o projeto, foi realizado um levantamento dos diversos modos de operação existentes no ESP32, o qual ficará responsável por toda a parte do condicionamento dos sinais dos sensores e parte da comunicação sem fio. Para exemplificar, existem dois modos extremos de consumo: processamento e transmissão de dados constante onde a todo momento há interação de dados com o *smartphone*; ou transmissões momentâneas e intervaladas onde o maior consumo concentra-se em um momento particular para transmissão de dados. Como no monitoramento de sono não há necessidade de observar os sinais vitais em tempo real, também se abre oportunidades para transmissão apenas na finalização das leituras, momento em que ele acorda, ou cada minuto, por exemplo. Verificando o manual do ESP32, observa-se que o consumo para *Bluetooth* de baixo consumo (BT/BLE) não ultrapassa os 130mA.

Tabela 5- Tabela retirada do manual do ESP32 referente a consumacao proporcional aos seus estados de transmissão sem fio.

Fonte: Manual da linha ESP32.

Mode	Min	Typ	Max	Unit
Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	240	-	mA
Transmit 802.11b, OFDM 54 Mbps, POUT = +16 dBm	-	190	-	mA
Transmit 802.11g, OFDM MCS7, POUT = +14 dBm	-	180	-	mA
Receive 802.11b/g/n	-	95 ~ 100	-	mA
Transmit BT/BLE, POUT = 0 dBm	-	130	-	mA
Receive BT/BLE	-	95 ~ 100	-	mA

Para estimar a capacidade necessária da bateria para alimentar o sistema, utilizou-se a tabela de consumo da Tabela 5. Assim, estimou-se que em metade do tempo total de operação o ESP32 encontra-se em transmissão de dados, enquanto na outra metade, em recebimento. Assim, o consumo médio de corrente do sistema é igual à média de corrente destes dois estados. Dessa forma, para calcular a capacidade estimada da bateria, basta analisar o produto entre o consumo médio e a autonomia total requerida. Neste trabalho, foi estipulado que o tempo total de operação deve ser de aproximadamente 10 horas.

$$C_{est} = (0,5i_{tx} + 0,5i_{rx})A$$

Equação 1 - Equação para a capacidade estimada da bateria.

Desta forma, utilizando a Equação 1, a capacidade estimada da bateria deve ser de, pelo menos, 1150 mAh. Porém, adicionando-se um fator de segurança e considerando-se o consumo dos periféricos do ESP32, concluiu-se que uma bateria de 1800 mAh é um valor razoável para o primeiro protótipo, tendo em vista que há sobra de energia para a prototipagem. Contudo, conforme melhorias de consumo e eficiência forem implementadas, esta capacidade pode certamente diminuir de forma considerável.

Usando a bateria de lítio-polímero com a capacidade definida, cuja imagem está na Figura 35, juntamente ao microcontrolador e um BMP180, foram feitas três transmissões por segundo dos dados de pressão atmosférica local e a temperatura da própria bateria. As transmissões foram direcionadas ao aplicativo *Android* que recebeu e armazenou toda leitura em um arquivo de texto.



Figura 35 - Imagem da bateria lítio-polímero de 1800mAh usada para suportar o processamento do ESP com a transmissão de dados.
Fonte: Os Autores.

Como foi aplicada uma coleta constante e transmissão a cada três segundos, foi atribuído um consumo maior que o casual, viabilizando um teste empírico. Inicialmente a bateria foi carregada por completa e colocada em um conversor AD em paralelo com o ESP32. O teste começou com a bateria completamente carregada e terminou quando a transmissão foi interrompida naturalmente por falta de capacidade oferecida ao sistema. O diagrama do teste pode ser observado na Figura 36.

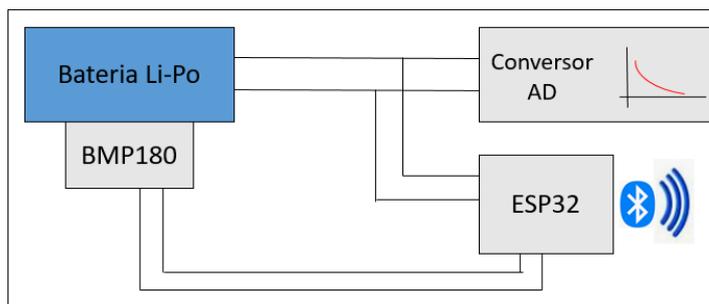


Figura 36 - Esquemático mostrando o conversor AD coletando a tensão de descarga da bateria.
Fonte: Os Autores.

O resultado foi positivo, pois funcionou corretamente por nove horas até parar de transmitir - tempo suficiente para fazer uma coleta noturna. Como nesse sistema, o ESP enviou três amostras por segundo, e no sistema projetado se estipula uma transmissão a cada 5 segundos aproximadamente, há garantia que o consumo será menor ao decorrer da coleta, por consequência uma autonomia maior que as nove horas no projeto finalizado. O aquecimento da bateria não se mostrou problema, pois apesar de exigir consumo constante, a temperatura coletada na bateria mostrou-se equivalente à temperatura ambiente durante todo período de uso. O gráfico da Figura 37 demonstra a descarga durante o teste.

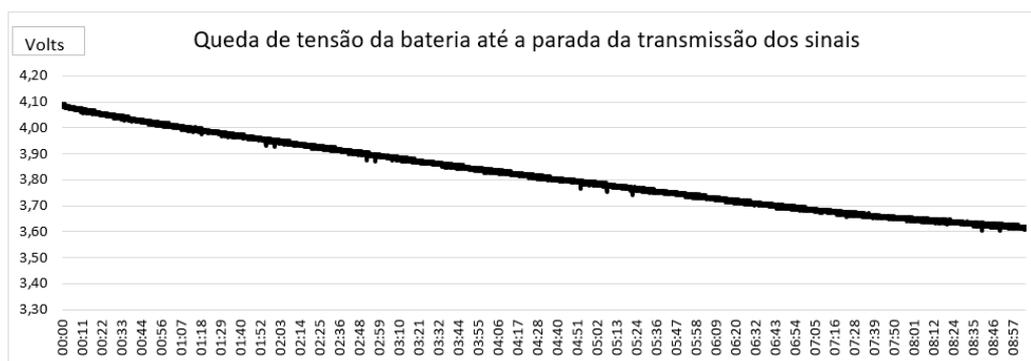


Figura 37 - Monitoramento da descarga da bateria do sistema de transmissão e processamento de dados do ESP32.
Fonte: Os Autores.

4 RESULTADOS

O resultado proposto deste projeto foi, através do sistema de sensoriamento desenvolvido, avaliar o comportamento de sinais característicos do sono em um usuário ao decorrer das horas de coleta. Para atingir este resultado principal, foram implementadas duas etapas fundamentais: a primeira é a criação de sensores específicos para leitura das grandezas do sono (frequência respiratória, batimento cardíaco, temperatura e bruxismo); a segunda etapa envolveu a comunicação desses sinais para o sistema operacional *Android*. Através do módulo ESP 32, os dados foram enviados ao *smartphone* que fez o tratamento e exibição destes dados numa interface gráfica para o usuário.

O sensor de bruxismo não foi viabilizado para rede de sensores devido ao seu modo construtivo. Para confeccioná-lo foi necessário adquirir placas de silicone que se moldam no formato da arcada dentária do usuário, logo seu formato torna-se um padrão complicado de se manter. Também, apesar de ter uma construção rústica, acabou se tornando muito intrusivo devido ao fio que transmite os dados da pressão da mordida e ao mesmo tempo alimenta o sensor no interior das placas de silicone. Logo, apesar de ter bons resultados com a detecção dos sinais de bruxismo, os pontos críticos deixaram o sensor afastado da rede definitiva. O algoritmo para detecção de apneia aplicado no sensor de frequência respiratória também não apresentou resultados condizentes devido aos ruídos que foram intrusivos na parada respiratória, segundo os testes com os autores. Para isso teria que sensibilizar mais ainda o sensor de frequência respiratória, que foi possível dentro do prazo estipulado para o projeto, pois envolveria mais análises de sinais e um algoritmo mais estruturado sendo que em paralelo estavam sendo desenvolvidos outros sensores e funções para comunicação sem fio. Entretanto, apesar dos desafios descritos, foi possível retirar conclusões interessantes com os dados vitais coletados pelos sensores.

4.1 RESULTADO DAS COLETAS DOS SENSORES E CORRELAÇÕES

Para validar os sensores por completo, foram realizadas coletas noturnas com os próprios autores a fim de verificar pontos críticos e melhorias nos protótipos. Como o Raspberry Pi não possui portas analógicas, foi utilizado para receber os valores de

movimentação noturna, enquanto em paralelo o Arduino deu suporte com o sensor de batimento cardíaco e frequência respiratória. Assim foi feita a coleta mais adequada dos sinais do batimento cardíaco médio (Figura 38), frequência respiratória (Figura 39), em paralelo com o sensor de movimentação noturna (Tabela 6).

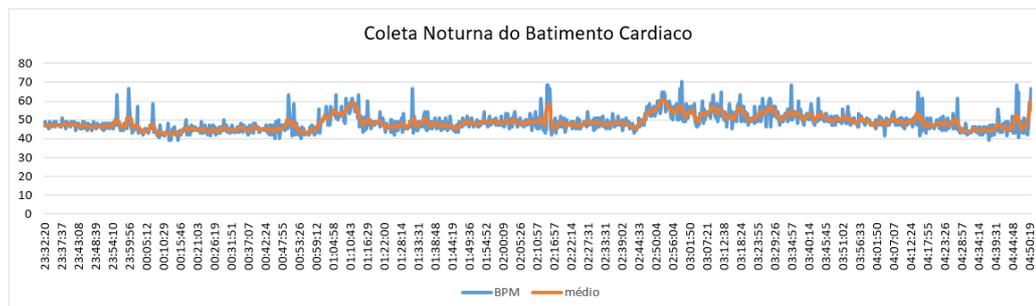


Figura 38 - Resultado do monitoramento noturno do batimento cardíaco
Fonte: Os autores.

Apesar de alguns picos surgirem durante a noite, a média total do batimento cardíaco ficou abaixo dos 50 bpm, o que condiz com as bibliografias. Salvo a média geral, alguns picos fizeram a média subir um pouco em um período específico, próximo à uma hora da manhã, onde houve um aumento para 60 bpm. Assim como nos batimentos cardíacos, o mesmo comportamento pode ser observado na frequência respiratória média.

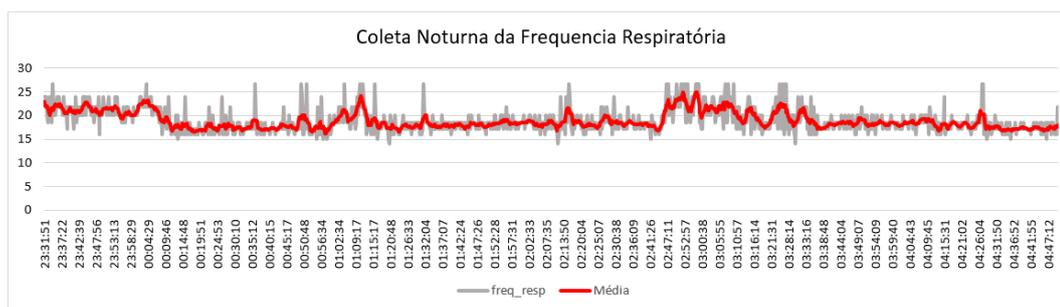


Figura 39 - Resultado do monitoramento noturno da frequência respiratória.
Fonte: Os autores.

Os resultados dos sinais coletados da frequência respiratória, assim como dos batimentos cardíacos, ficaram numa faixa correspondente às bibliografias (entre quinze e vinte irms - índice de respiração média). Inicialmente, ao colocar a faixa torácica do sensor o autor ainda estava acordado, marcando uma respiração média de 20 irms, e após meia noite o irms cai para 16 a 18. Pode-se concluir que a diminuição da frequência respiratória nesse ponto teve relação com a atividade corpórea, e que

a partir de meia noite há um relaxamento corporal. Esse relaxamento pode ser observado pelo sensor de movimentação noturna que indica um período ausente de movimentos entre meia noite e duas e quinze da manhã como pode ser observado na Tabela 6.

Tabela 6 - Período de sono profundo detectado pelo sensor de movimentação noturna.
Fonte: Os autores

Horário	Intensidade do Movimento	movimento(acumulado)	temperatura
23:59:37	1,62	31	33,2
23:59:42	1,17	32	33,2
23:59:47	1,43	33	33,2
02:15:10	1,02	34	37,1
02:15:15	8,31	35	36,9

Com o sensor de movimentação noturna foi possível verificar se os ruídos captados pelos sensores do corpo são resultados de movimentações noturnas. Como não teve registro de movimentos, pode-se concluir que o aumento da frequência respiratória e do batimento cardíaco podem ser resultados de alguma atividade noturna como período REM (rapid eye movement - Movimento rápido ocular).

O REM tem esse nome devido ao evento que o caracteriza. Segundo FELSON, 2018, é um fenômeno que ocorre na etapa mais profunda do sono onde toda conexão muscular é desligada do cérebro - menos os músculos dos olhos (que entram em movimentos acelerados). Nesse estágio, os sinais da frequência respiratória sobem e os batimentos do coração aceleram, dando origem ao sonho, justamente como foi detectado pelos sensores desenvolvidos.

Outra análise feita para verificar a veracidade do aumento da frequência respiratória, para descartar ruídos externos naquele período da madrugada, é avaliar o seguimento da temperatura do sensor contido no invólucro de ar da cinta torácica (embarcado no BMP180).

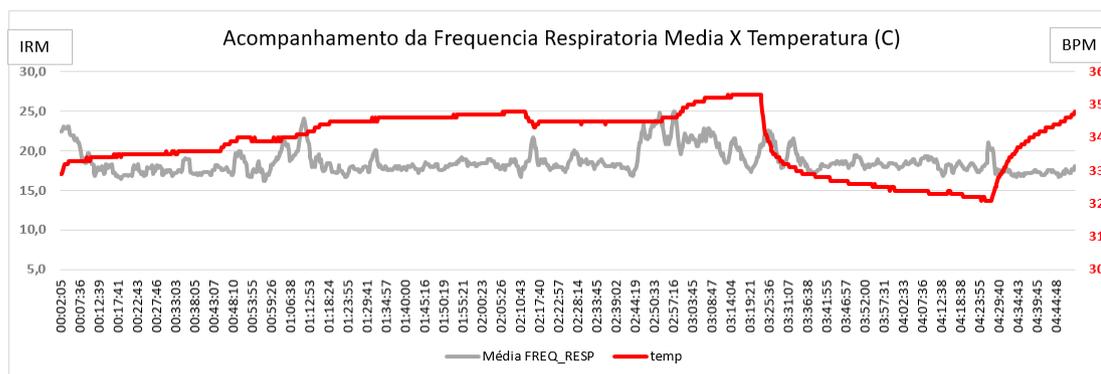


Figura 40 - Acompanhamento da temperatura lida no interior do invólucro de ar da cinta toracia em paralelo com a respiração média.
Fonte: Os autores.

Seguindo a escala de temperatura do gráfico da Figura 40, no período da uma hora da manhã, não há diferença abrupta na temperatura lida pelo sensor de frequência respiratória - ficando numa faixa de 34 a 35 graus Celsius. Após três da manhã, onde há uma oscilação significativa na frequência respiratória, a temperatura coletada pelo transdutor cai de 35 para 33, levantando umas hipóteses de que aconteceu um desacoplamento do sensor no tórax ou descoberta do local - deixando-o exposto ao frio da noite.

Outra análise retirada dos resultados dos sinais lidos pelos sensores, foi a correlação entre a frequência respiratória e o batimento cardíaco no período atribuído ao REM durante a coleta com os autores conforme mostrado na Figura 41.

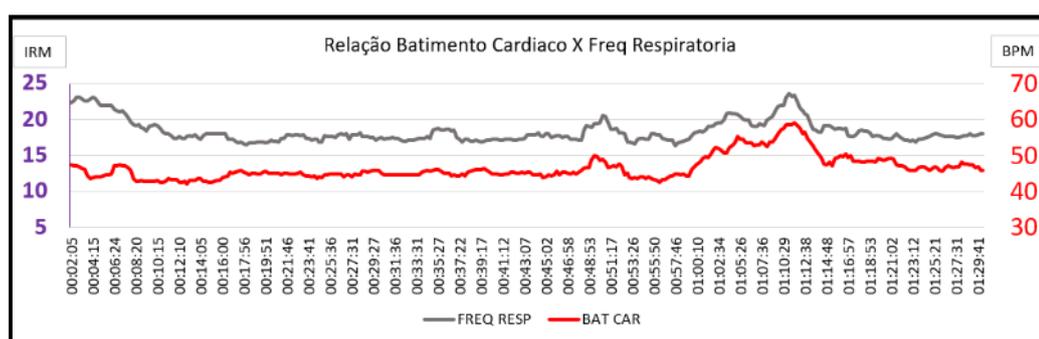


Figura 41 – Resultado da relação da frequência respiratória com os batimentos cardíacos médios no período de sono profundo e REM.
Fonte: Os autores.

Normalmente o sono REM acontece noventa minutos após pegar no sono. Na primeira sequência (primeiro estágio) ele dura pelo menos dez minutos, nesse período os batimentos cardíacos e a frequência respiratória aumentam. Existe uma sequência de ocorrência do REM ao longo do sono, e os últimos estágios duram até uma hora

(FELSON, 2018). Analisando o gráfico acima, pode-se dizer que o autor pegou no sono a partir da meia noite (leva-se em consideração também a ausência de movimentos noturnos). Aproximadamente sessenta minutos após o sono, onde o autor está em sono profundo, os batimentos e a frequência respiratória aumentam. O aumento dessas taxas duram em torno de dez minutos. Logo, pode-se concluir que o autor estava passando pelo período de sono REM. Interessante notar também, que na correlação da frequência respiratória e taxa cardíaca, os sinais são muito semelhantes na grande maioria do tempo - concluindo então a total dependência da oxigenação do sangue com a respiração do usuário.

4.2 INTERFACE GRÁFICA DO APLICATIVO

Tratando-se da interface gráfica do aplicativo, a sequência de atividades mostrou-se intuitiva se o nome do ESP32, exibido na lista da atividade de busca, é conhecido.

Ao final da coleta de dados, clicando no botão “Parar”, os resultados são exibidos na tela. O usuário pode, então, observar os pontos críticos da noite de sono apenas observando os gráficos e resultados no *smartphone*.

De qualquer forma, ao coletar os dados pelo documento de texto salvo, é possível que o usuário realize uma análise mais aprofundada, possivelmente com o auxílio de um profissional da saúde habilitado.

Na Figura 42 é possível observar a atividade de controle do dispositivo conectado do aplicativo exibindo os dados coletados durante uma hora de teste. Nesta tela, é possível observar os gráficos dos valores lidos, os quais aceitam aproximação e rolagem na tela com o movimento de pinça dos dedos, funcionalidade útil para observar pequenas variações em longas horas de coleta. Também são visíveis os resultados de valores máximos, mínimos e médios das frequências respiratória e cardíaca, temperatura e intensidade de movimento.

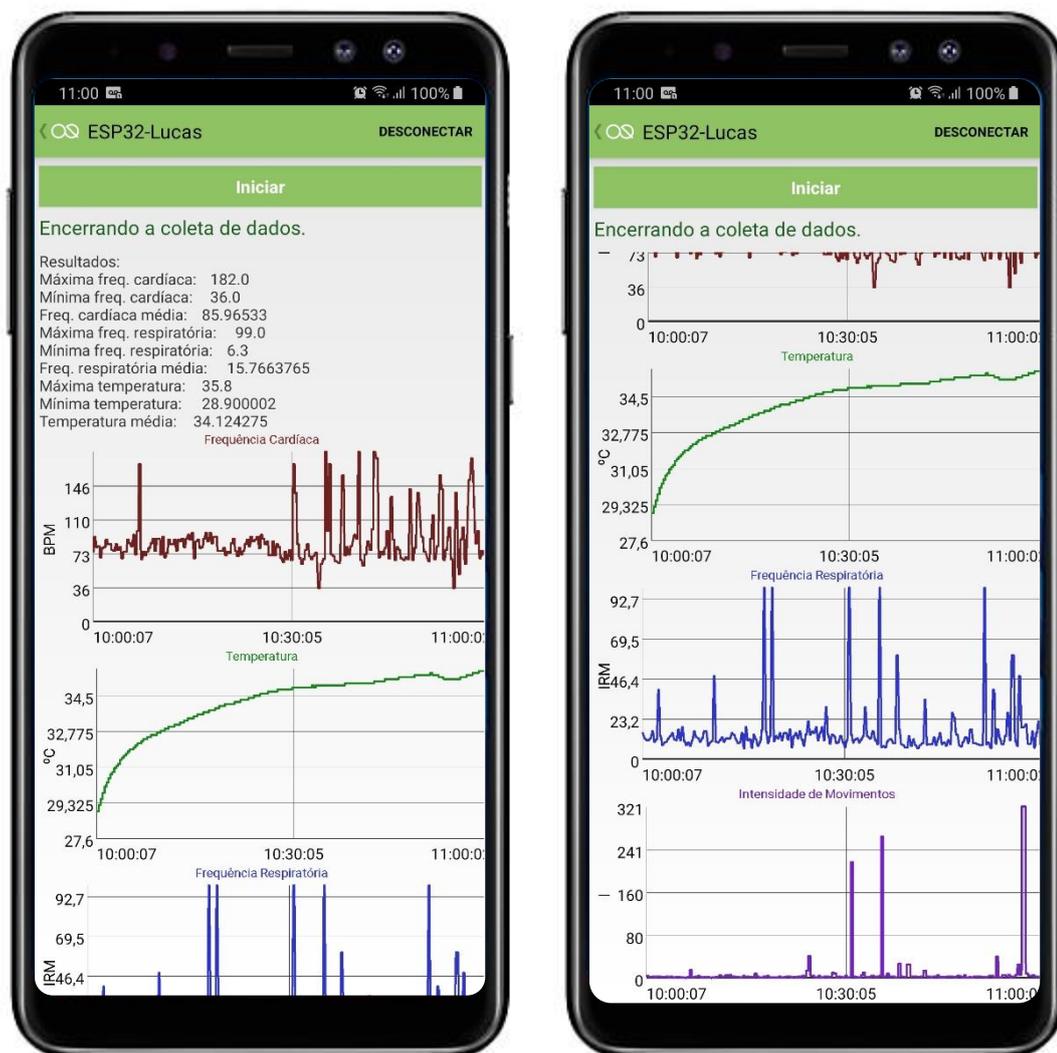


Figura 42 - Atividade do aplicativo exibindo os resultados dde uma coleta de teste.
Fonte: Os Autores.

4.3 DISCUSSÃO

A auto quantificação prevê a captação dos sinais biológicos lidos pelos sensores desenvolvidos mostrando correlações entre si. Assim, podem ser avaliados padrões comportamentais e futuramente desenvolver sistemas mais abrangentes para coleta de dados e aprimorar mais ainda a auto quantificação na área em que se aplica. Este projeto abre possibilidades para análises mais aprofundadas do sono em trabalhos futuros e testes em grupos. Os trabalhos futuros incluem melhorias na sensibilidade dos sensores como o de frequência respiratória a fim de detectar com mais exatidão as ocorrências e causas da apneia; implementações de outras variáveis no sistema como curvas de temperatura corpórea distribuída; expansão nos pontos de movimentos noturnos para detectar um estado completo de sono profundo; melhoria na interface gráfica para melhor realimentação dos dados vitais ao usuário, etc. Os testes em maior número de pessoas podem auxiliar na validação das relações entre as variáveis medidas e a qualidade do sono, níveis de estresse e correlacionar todos os dados comportamentais do dia com a qualidade do descanso do usuário. A Figura 43 apresenta características do protótipo desenvolvido.

Equipamento	Sensores	Preço	Portabilidade	Conectividade	Confiabilidade
Protótipo deste trabalho	<ul style="list-style-type: none"> • Frequência cardíaca • Movimentação noturna • Temperatura • Frequência respiratória 	Custo do material para o protótipo R\$ 150	Faixa torácica	<ul style="list-style-type: none"> • USB • BLE 	Na versão atual, os resultados são aproximados

Figura 43 – Tabela de características do protótipo.
Fonte: Os Autores.

5 CONCLUSÃO

Foi desenvolvido um sistema composto de sensores capazes de ler sinais vitais durante o sono. Os sensores foram testados individualmente e suas leituras físicas bem caracterizadas. Com o sensor de pressão entre os dentes foi possível detectar os sinais relacionados ao bruxismo. Com o sensor de batimento cardíaco óptico em conjunto com o sensor de frequência respiratória, foi possível monitorar uma noite de sono completa e observar correlação entre ambos sinais registrados. O sensor de movimentação noturna teve uma importância grande no projeto, pois com ele foi possível mostrar, durante a noite de coleta, períodos que o usuário mostrou ausência de movimento. Neste período, com a sobreposição dos sinais de frequência respiratória e batimento cardíaco, tem-se uma forte indicação do sono REM (sono profundo), e também de interrupções do sono profundo de acordo com a intensidade dos movimentos que o autor realizou. Também pode ser citada a utilidade do sensor de temperatura junto à cinta de frequência respiratória que serviu como variável de validação dos sinais captados pelo sensor da faixa torácica, evidenciando quando ela se desloca do corpo. Desta forma, os sensores foram utilizados de forma integrada no monitoramento de sono, resultando num sistema para aquisição dos próprios sinais vitais assim como definido no campo da auto quantificação.

O ESP32 teve desempenho adequado para esta aplicação por vários motivos. Isso pode ser afirmado por três principais motivos. O poder de processamento do SoC foi suficiente para realizar com agilidade todas as rotinas atribuídas a ele, desde a leitura e condicionamento dos valores lidos dos sensores, até a transmissão com o protocolo de comunicação e gerenciamento do sistema BLE. O consumo de energia requerida pelo dispositivo permitiu que ele fosse alimentado por uma bateria cuja capacidade é razoável, tanto em termos de comparação com *smartphones* quanto em preço de mercado. O sistema atingiu todas as expectativas quanto à capacidade de comunicar-se via BLE. O módulo de desenvolvimento foi excelente para este trabalho, tendo em vista que, além de suportar todos os requerimentos possui capacidade para atender mais funcionalidades que podem ser implementadas futuramente.

5.1 TRABALHOS FUTUROS

Levando em consideração os pontos críticos do sensor de bruxismo, pode-se aprimorá-lo para realização de medidas de forma menos intrusiva. Uma alternativa seria embarcar na prótese um sistema de alimentação de baixo consumo e transmissão sem fio. Na ocorrência de ranger de dentes, pode ser enviada informação ao microcontrolador para aumentar a atenção aos sensores priorizando coleta do batimento cardíaco e frequência respiratória mais detalhados. Esta ideia de priorização veio em função dos resultados que este projeto mostrou: correlações entre os sinais. Será que existe alguma relação do bruxismo com os sinais vitais do usuário? Qual seria o momento da ocorrência do bruxismo em relação ao período de sono da pessoa? Essas questões despertaram nos autores curiosidades que podem ser levadas a frente para novas pesquisas. Outra possibilidade de expansão é fazer um monitoramento mais detalhado dos movimentos corpóreos. A coleta pode ser expandida para os diversos membros do usuário como braços, pernas e cabeça. Assim pode ser verificado se os movimentos corpóreos estão presentes durante todo estágio de sono ou acumulados em um período específico da noite.

O sensor de frequência respiratória pode ser aprimorado para se obter sensibilidade maior. Uma possibilidade seria colocar vários sensores de pressão dentro de outros invólucros de ar, assim um ciclo respiratório ruidoso por um pode ser compensado por outras perspectivas. Também, visto que sua aplicação é vasta, implementar um algoritmo mais eficaz para detectar os casos de apneia. Neste caso, seria interessante usar a detecção de apneia como gatilho para ativar uma leitura mais detalhada dos sensores de batimento cardíaco - que está relacionado à oxigenação do sangue e por consequência à saúde do usuário.

Referente ao sistema de aquisição, podem ser utilizados os microfones do próprio celular como sensores de ronco. O ronco está ligado à respiração, e por consequência à frequência respiratória. Ou seja, pode ser avaliado qual o nível de ronco e correlacionar estes dados com a respiração e oxigenação do sangue (através do sensor óptico dos batimentos cardíacos). Para isso, deve ser avaliada a viabilidade tendo em conta processamento digital de sinais. Pois apesar do ronco ser perceptível aos ouvidos humanos, em termos de eletrônica não é diferente de ruídos do ambiente (como cachorro latindo ou caminhões passando pela rua durante a noite). Logo devem

ser levados em consideração filtros digitais ou analógicos para eliminar ou aliviar esses ruídos.

Sensores de temperatura, apesar de existirem embarcados nos sensores de movimentação noturna e frequência respiratória, não são voltados a coletar temperatura corpórea. Entretanto, é interessante voltar a atenção para esse tipo de coleta, pois durante o sono REM ou sono profundo, segundo biografias citadas acima, a temperatura corporal tende a cair. Logo é uma ótima variável para confirmar o estado do sono do usuário.

REFERÊNCIAS

BRUNA, Maria Helena Varella. **Bruxismo**. Disponível em: <https://drauziovarella.uol.com.br/doencas-e-sintomas/bruxismo-2/>. Acesso em: 7 de fevereiro de 2019.

CALVAZERA, Bruno. **O que é tecnologia autoquantificada, 2013**. Disponível em: <https://hypescience.com/tecnologia-autoquantificada/>. Acesso em: 08 de Fevereiro de 2019

CYMBALUK, Fernando. **O que acontece quando dormimos menos de oito horas?** Disponível em: <https://noticias.uol.com.br/saude/ultimas-noticias/redacao/2017/01/30/dividas-com-o-proprio-sono-causam-diversos-problemas-de-saude.htm>. Acesso em: 12 de fevereiro de 2019.

DE LUCENA, Samuel E. et al. ECG monitoring using *Android* mobile phone and *Bluetooth*. In: **Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International**. IEEE, 2015. p. 1976-1980.

DEV, Harsh et al. *Android* Based Home-Automation using Microcontroller. **International Journal of Advanced Networking and Applications**, v. 10, n. 1, p. 3724-3727, 2018.

DIGIKEY. **ESP32-WROOM-32 Espressif Systems | RF/IF and RFID | DigiKey**. Disponível em: <https://www.digikey.co.il/product-detail/en/espressif-systems/ESP32-WROOM-32/1904-1010-1-ND/8544305>. Acesso em: 7 de fevereiro de 2019.

ESPRESSIF. **ESP32 Overview**. Disponível em: <https://www.espressif.com/en/products/hardware/esp32/overview>. Acesso em: 7 de fevereiro de 2019.

ESPRESSIF. **ESP32 WROOM Series Overview**. Disponível em: <https://www.espressif.com/en/products/hardware/esp-wroom-32/overview>. Acesso em: 7 de fevereiro de 2019.

FIOCRUZ. **Dormir bem é fundamental para a saúde, alertam especialistas**. Disponível em: <https://portal.fiocruz.br/noticia/dormir-bem-e-fundamental-para-saude-alertam-especialistas>. Acesso em: 6 de fevereiro de 2019.

GARTNER. **Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017**. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2018-02-22-gartner-says-worldwide-sales-of-smartphones-recorded-first-ever-decline-during-the-fourth-quarter-of-2017>. Acesso em: 6 de fevereiro de 2019.

GUYTON, Arthur C; HALL Jhon E. Tratado de Fisiologia Médica, v.11, p. 478, 2006.

IBGE. **PNAD: Acesso à Internet e à Televisão e Posse de Telefone Móvel Celular para Uso Pessoal 2016**. Disponível em:

https://agenciadenoticias.ibge.gov.br/media/com_mediaibge/arquivos/c62c9d551093e4b8e9d9810a6d3bafff.pdf. Acesso em: 6 de fevereiro de 2019.

International Data Corporation. **Após dois anos, mercado de smartphones cresce em 2017 e atinge o segundo melhor desempenho de vendas.** Disponível em: <http://br.idclatin.com/releases/news.aspx?id=2312>. Acesso em: 12 de fevereiro de 2019.

KUMAR, Maradugu Anil; SEKHAR, Y. Ravi. *Android based health care monitoring system.* In: **Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on.** IEEE, 2015. p. 1-5.

LIMA, Ana L. **Qual a Frequência Cardíaca normal, alta e baixa e como normalizar.** Disponível em: <https://www.tuasaude.com/frequencia-cardiaca/>. Acesso em: 8 de fevereiro de 2019.

MAGAGNIN JUNIOR, Ari; SIMAS, João Hamilton Cecato. **Eletrocardiógrafo com dispositivos móveis e monitoramento a distância.** 2013. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

PANDYA, Bhavik et al. *Android based home automation system using Bluetooth & voice command.* **International Research Journal of Engineering and Technology (IRJET)**, v. 3, n. 03, 2016.

ROCHA, Maria Cecília Pires da; MARTINO, Milva Maria Figueiredo De. Stress and sleep quality of nurses working different hospital shifts. **Revista da Escola de Enfermagem da USP**, v. 44, n. 2, p. 280-286, 2010.

VEIGA, Jeangrei et al. Aplicações móveis com interação médico-paciente para um estilo de vida saudável: uma revisão sistemática. 2017.