



**UNIVERSIDADE
FEDERAL
DE PERNAMBUCO**

Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Departamento de Eletrônica e Sistemas

Graduação em Engenharia Eletrônica

Pedro Campello Tôrres de Azevedo Teles

**Sistema Biofeedback wireless eletromiográfico
para auxílio em tratamentos musculares da região
pélvica**

Recife
Setembro de 2023

Pedro Campello Tôrres de Azevedo Teles

**Sistema Biofeedback wireless eletromiográfico
para auxílio em tratamentos musculares da região
pélvica**

Trabalho de Conclusão apresentado ao Curso de Graduação em Engenharia Eletrônica do Departamento de Eletrônica e Sistemas da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia Eletrônica.

Orientador: Prof. Hermano Andrade Cabral, Ph.D.

Recife

Setembro de 2023

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Teles, Pedro Campello Tôrres de Azevedo.

Sistema Biofeedback wireless eletromiográfico para auxílio em tratamentos
musculares da região pélvica / Pedro Campello Tôrres de Azevedo Teles. -
Recife, 2023.

48p : il., tab.

Orientador(a): Hermano Andrade Cabral

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de
Pernambuco, Centro de Tecnologia e Geociências, Engenharia Eletrônica -
Bacharelado, 2023.

1. Biofeedback. 2. Eletromiografia. 3. ESP32. I. Cabral, Hermano Andrade.
(Orientação). II. Título.

600 CDD (22.ed.)

Pedro Campello Tôrres de Azevedo Teles

**Sistema Biofeedback wireless eletromiográfico
para auxílio em tratamentos musculares da região
pélvica**

Trabalho de Conclusão apresentado ao Curso de Graduação em Engenharia Eletrônica do Departamento de Eletrônica e Sistemas da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia Eletrônica.

Banca Examinadora

Prof. Hermano Andrade Cabral, Ph.D.
Universidade Federal de Pernambuco

Prof. Guilherme Nunes Melo, D.Sc.
Universidade Federal de Pernambuco

Recife

Setembro de 2023

*Dedico este trabalho a todos que
me ajudaram, apoiaram e se
fizeram presentes durante esse
trajeto.*

Agradecimentos

Aos meus pais, irmão e a toda minha família que durante todo o processo acreditaram e me apoiaram para que eu completasse esta etapa tão importante de minha vida. Aos amigos que fiz durante a graduação, em especial Pedro Félix, Letícia, Vagner e Poliana. Sem eles não seria capaz de chegar aqui da forma que cheguei.

A todos os professores do Departamento de Eletrônica e Sistemas, mas principalmente ao professor e orientador deste trabalho Hermano Cabral, que de forma paciente e solícita me auxiliou da melhor forma possível sempre que precisei de sua ajuda para realizar este projeto.

Por último, gostaria de agradecer de forma especial a minha esposa e companheira, Beatriz Dornelas, que durante todo o processo não mediu esforços para me ajudar das mais diversas maneiras, frente a todos os desafios que enfrentei.

Para nós os grandes homens não
são aqueles que resolveram os
problemas, mas aqueles que os
descobriram

Albert Schweitzer

Resumo do Trabalho de Conclusão de Curso apresentado ao DES/UFPE como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Eletrônica (Eng.)

Sistema Biofeedback wireless eletromiográfico para auxílio em tratamentos musculares da região pélvica

Pedro Campello Tôrres de Azevedo Teles

A disfunção pélvica afeta a vida de inúmeras pessoas, e ocorre quando os músculos do assoalho pélvico não funcionam de forma devida, provocando a contração ou relaxamento excessivo da musculatura. Essa condição fisiológica pode gerar além de problemas físicos, consequências mentais e sociais. A técnica de Biofeedback é uma poderosa aliada na intervenção da disfunção do assoalho pélvico (DAP), principal responsável por condições como incontinência urinária, prolapso genital e distúrbios sexuais. Entretanto muitos dos equipamentos utilizados em seu tratamento focam na utilização de uma sonda pressórica anal ou vaginal para extrair os estímulos, e esta técnica pode gerar certo desconforto. Em virtude disso, o trabalho propõe um equipamento eletrônico que se utilize de biofeedback e eletromiografia para auxílio de tais enfermidades.

Palavras-chave: Biofeedback; Eletromiografia; ESP32 .

Abstract of Course Conclusion Work presented to DES/UFPE as a partial fulfillment of the requirements for the degree of Bachelor of Electronic Engineering (Eng.)

Electromyographic wireless Biofeedback system to aid in muscle treatments of the pelvic region

Pedro Campello Tôrres de Azevedo Teles

Pelvic dysfunction affects the lives of countless people, this occurs when the pelvic floor muscles do not work properly, causing excessive muscle contraction or stretching. The mentioned dysfunction can generate, in addition to physical problems, psychological problems. The Biofeedback technique is a powerful ally in the intervention of pelvic floor dysfunction, primarily responsible for conditions such as urinary incontinence, genital prolapse and sexual disorders. However, most of the equipment used in its treatment focus on the use of an anal or vaginal pressure probe to extract the stimuli, yet this technique can cause some discomfort. The work shows an electronic equipment that uses biofeedback and electromyography to help with such illnesses.

Keywords: Biofeedback; Electromyography; ESP32

Sumário

Lista de Figuras	xii
------------------	-----

Lista de Abreviações	xiv
----------------------	-----

1	Introdução	1
1.1	Objetivo Geral	2
1.1.1	Objetivos específicos	2
1.2	Organização do TCC	3
2	Fundamentação Teórica	5
2.1	Músculos do Assoalho Pélvico	5
2.1.1	Tratamentos das disfunções	7
2.2	Eletromiografia e Eletrocardiografia	8
2.3	Bluetooth Low Energy	9
2.4	ESP32 DEVKITV1	12
2.5	Módulo ECG	13
2.6	Conversor <i>Buck</i>	16
2.7	Python	19
2.7.1	Bleak	19
2.7.2	Matplotlib	20
2.7.3	Asyncio	20
2.7.4	Pandas	20

3	Desenvolvimento	22
3.1	Sistema Proposto	22
3.1.1	Aquisição do Sinal	23
3.1.2	Circuito de Alimentação	24
3.1.3	Processamento do Sinal	26
3.1.4	Firmware	28
3.1.5	Software	31
4	Resultados	35
4.1	Hardware	35
4.2	Envio e Armazenamento dos Dados	36
4.3	Análise dos Gráficos	38
5	Considerações finais	42
5.1	Conclusão	42
5.2	Dificuldades Encontradas	42
5.3	Trabalhos Futuros	43
	Referências Bibliográficas	45

Lista de Figuras

1.1	Exemplo de aparelho de Biofeedback comumente utilizado.	2
2.1	Músculos do assoalho pélvico.	6
2.2	Densidade espectral de potência EMG(Vermelho) - ECG(Azul). . . .	9
2.3	BLE pilha de protocolos.	10
2.4	Hierarquia de encapsulamento do objeto GATT.	12
2.5	ESP32 pinout.	13
2.6	Módulo monitor cardíaco AD8232.	14
2.7	Circuito ECG AD8232.	15
2.8	Gráfico da magnitude da resposta em frequência do módulo monitor cardíaco.	16
2.9	Circuito Buck.	17
2.10	Módulo Step Down LM2596.	18
2.11	Módulo DC-DC LM2596.	19
2.12	Demonstração de fluxo de código usual vs assíncrono.	20
3.1	Diagrama geral.	23
3.2	Eletrodos posicionados sobre musculatura do antebraço.	23
3.3	Posicionamento correto dos eletrodos para uma aplicação de Biofe- edback da região pélvica.	24
3.4	Esquemático do circuito de alimentação simétrica.	25
3.5	Circuito de alimentação simétrica na protoboard.	25
3.6	Esquemático do circuito no LTSpice.	27

3.7	Resultado de análise AC realizada no LTSpice.	28
3.8	Classes com métodos de callback.	29
3.9	Trecho do código com função setup.	30
3.10	Trecho do código com função loop.	31
3.11	Trecho do código fonte da classe <i>Connection</i>	32
3.12	Trecho do código fonte da classe <i>DataToFile</i>	33
3.13	Trecho do código fonte da função <i>Main</i>	34
4.1	Protótipo do circuito em PCB.	36
4.2	Exemplo de arquivo CSV gerado pelo software ao final de uma seção.	38
4.3	Captura de tela demonstrando o resultado observado na aplicação Python, utilizando o módulo ECG AD8232.	40
4.4	Captura de tela demonstrando o resultado observado na aplicação Python, utilizando o circuito EMG desenvolvido.	41

Lista de Abreviações

MAP	Músculos do Assoalho Pélvico
DAP	Disfunções do Assoalho Pélvico
OMS	Organização Mundial da Saúde
EMG	Eletromiografia
sEMG	Eletromiografia de superfície
ECG	Eletrocardiograma
Hz	Hertz
BLE	Bluetooth Low Energy
IoT	Internet of Things
GAP	Generic Access Profile
GATT	Generic Attribute Profile
UUID	Universally Unique Identifier
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
GPIO	General Purpose Input/Output
LCD	Liquid Crystal Display
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
DC	Direct Current
CC	Corrente Contínua
CSV	Comma Separated Values
IO	Input/Output
GND	Ground

AMP-OP	Amplificador Operacional
PA	Passa-alta
PB	Passa-baixa
PF	Passa-faixa
LCD	Liquid Crystal Display

Capítulo 1

Introdução

DISFUNÇÃO do assoalho pélvico é uma condição clínica que acomete homens e principalmente mulheres, trazendo diversas consequências. Um exemplo é a incontinência urinária, vista como um problema de saúde pública pela Organização Mundial de Saúde, que aflige mais de 200 milhões de pessoas globalmente. No Brasil, estima-se que em torno de 11 a 23% da população feminina apresenta incontinência urinária e esse número pode chegar a 35% entre idosas [1].

Já existem diversas soluções consolidadas para auxiliar no tratamento, em sua maioria são dispositivos eletrônicos de *Biofeedback*, que almejam aferir a atividade muscular dos grupamentos da região pélvica e prover dados relevantes ao profissional de saúde durante as sessões. Muitos desses equipamentos focam na utilização de uma sonda pressórica anal ou vaginal para extrair os estímulos, porém esse método pode ser visto como algo invasivo e muitas vezes desmotiva a continuidade ou até o início do tratamento. Além disso, são aparelhos de alto valor comercial, muito grandes e com uma quantidade excessiva de fios, o que acaba por restringir seus cenários de uso [2].

Figura 1.1: Exemplo de aparelho de Biofeedback comumente utilizado.



Fonte: Fisioterapia Cavallazzi [3]

1.1 Objetivo Geral

O objetivo deste trabalho é apresentar uma alternativa de sistema eletrônico de baixo custo, compacta e sem fio que possa servir na análise e auxílio de tratamentos de disfunções musculares pélvicas. Para atingir esse objetivo, pretende-se utilizar algumas soluções de circuitos eletrônicos amplamente disponíveis, aliados a implementações de conceitos de engenharia de *hardware* e *software* aprendidos durante o curso de graduação.

1.1.1 Objetivos específicos

Para atingir os objetivos gerais, foram traçados tópicos específicos com metas e características almejadas para seguir durante o processo ou que estejam presentes no

resultado final do trabalho. As indicações a seguir levam em consideração o estado da arte, dispositivos e tecnologias já estabelecidos no mercado ou mencionados na literatura, para fins de comparação.

- Apresentar um protótipo de sistema embarcado (*hardware* e *software*) de relativa baixa complexidade, funcional e que cumpra o papel de um *Biofeedback* eletromiográfico;
- Manter um design voltado para portabilidade, com dimensões físicas reduzidas e emprego de comunicação *Wireless*;
- Utilizar componentes acessíveis, de baixo custo e fácil implementação, mantendo um modelo relativamente simples;
- Usar tecnologias de baixo consumo energético para garantir um melhor aproveitamento do tempo de uso da bateria;
- Descrever e documentar as técnicas e tecnologias utilizadas ao longo do trabalho;
- Exibir graficamente a resposta do sistema para que se possa analisar e extrair informações relevantes sobre a musculatura alvo.

1.2 Organização do TCC

O conteúdo deste TCC está dividido em cinco capítulos. As referências encontram-se nas páginas finais. A seguir, um resumo dos capítulos seguintes do TCC.

Capítulo 2. Introduz conceitos teóricos relevantes para o desenvolvimento deste trabalho.

Capítulo 3. Aborda o sistema proposto em uma visão geral e a forma como o projeto foi desenvolvido.

Capítulo 4. Expõe os resultados obtidos e conduz uma análise quanto ao que foi obtido em relação a proposta inicial.

Capítulo 5. Conclusão das ideias abordadas nos demais capítulos, ideias e propostas futuras, além de dificuldades encontradas.

Capítulo 2

Fundamentação Teórica

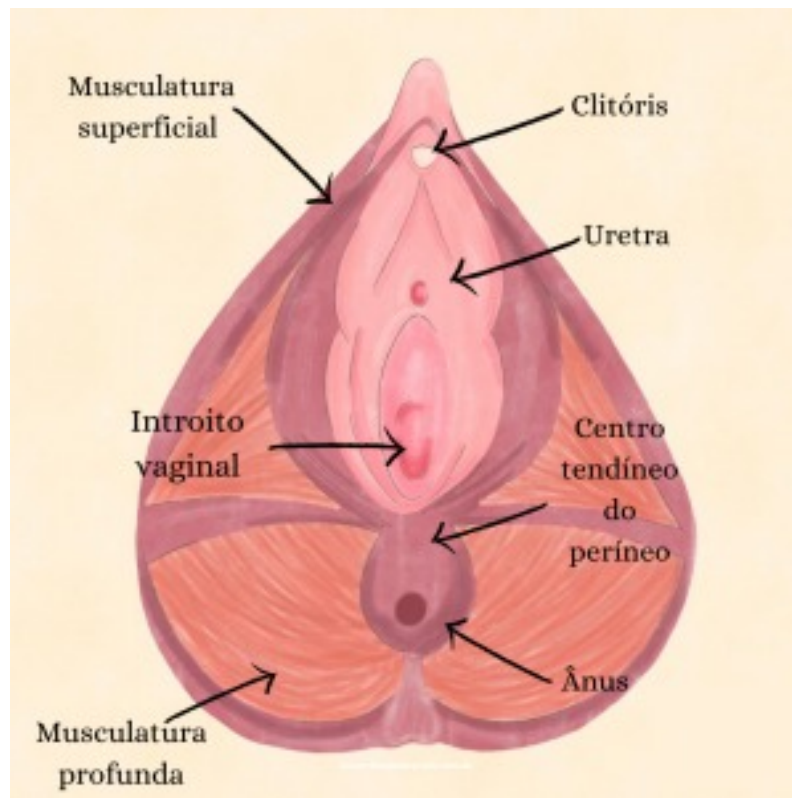
PARA o pleno entendimento da solução proposta e das ferramentas utilizadas, é necessário introduzir algumas ideias relacionadas ao problema, tecnologias e conceitos utilizados ao longo deste documento.

Serão abordados de forma breve alguns temas nas áreas de saúde, engenharia eletrônica, computação e afins, sendo apenas um compilado dos principais tópicos que possibilitaram o desenvolvimento do trabalho e por isso se mostram relevantes.

2.1 Músculos do Assoalho Pélvico

Como base para o entendimento fisiológico, é imprescindível esclarecer o que são os MAP e qual sua relevância no contexto deste trabalho. A sigla MAP significa Músculos/Musculatura do Assoalho Pélvico, que abrange os órgãos e musculatura da região baixa do abdômen. Assim como outros grupamentos musculares, parte de sua função fundamental está relacionada à sustentação dos órgãos da região abdominopelviana, além do auxílio em funções sexuais[4].

Figura 2.1: Músculos do assoalho pélvico.



Fonte: Lorem Corporis [5]

O mal funcionamento dessa musculatura pode desencadear uma série de problemas como incontinências, prolapsos, alterações evacuatórias, problemas sexuais, dentre outros, impactando a saúde e bem estar de milhões de pessoas. Fatores, como a alteração do peso corporal, complicações na gestação/parto, cirurgias na região pélvica, exercícios de alto impacto, podem dar início ou potencializar deficiências na função dos MAP. Essas complicações são conhecidas como Disfunções do Assoalho Pélvico (DAP) [6].

Apenas a incontinência urinária afeta mais de 200 milhões de pessoas no mundo, e de acordo com a Organização Mundial da Saúde (OMS) é considerada um problema de saúde pública [7]. No Brasil, estima-se que em torno de 11 a 23% da população feminina apresenta incontinência urinária e esse número pode chegar a 35% entre idosas [1].

2.1.1 Tratamentos das disfunções

Assim como outros músculos, os do assoalho pélvico podem ser exercitados e treinados para proporcionar uma melhor contração e coordenação, desempenhando dessa forma sua função de maneira plena e saudável [6].

Entre os procedimentos existentes para seu exercício destaca-se a cinesioterapia, que é um conjunto de movimentos realizados sob orientação do fisioterapeuta para fortalecer músculos. Por meio dos exercícios, busca-se melhorar os mecanismos de fechamento uretral, vesical e anal. Contudo, em razão da comum incapacidade dos pacientes em distinguir corretamente os músculos a serem exercitados, ocorrem contrações de outros músculos não relacionados, tais como o reto abdominal, glúteo máximo e adutor da coxa [6].

Outra modalidade de tratamento é a eletroestimulação. Neste recurso terapêutico os músculos são contraídos por meio de um aparelho que aplica diferentes impulsos elétricos a fim de criar um fortalecimento muscular. Para isso utiliza-se eletrodos intracavitários como meio de condução da corrente, por produzirem contração mais eficiente, já que há uma redução da interface entre o eletrodo e o músculo a ser contraído. Entretanto, existe certa resistência por parte das pacientes a esse tipo de eletrodo, em razão do desconforto relacionado a ele. Com a finalidade de causar menor incômodo ao paciente, é possível o uso de eletrodos de superfície, entretanto, neste caso a musculatura estimulada raramente é aquela que fará efeito em caso de incontinência urinária [6].

O tratamento por *Biofeedback* utiliza um aparelho capaz de obter informações da musculatura do assoalho pélvico e converter estes eventos em sinais visuais e/ou auditivos para o fisioterapeuta e o paciente. Este método é empregado como técnica para permitir ao indivíduo desenvolver a capacidade de controlar voluntariamente suas funções fisiológicas, pois tal tratamento consegue levar o paciente a associar os estímulos físicos, realizados durante a sessão de fisioterapia, aos sinais visuais obtidos pelo aparelho. Produzindo uma consciência de seus movimentos musculares, elevando sua força pélvica de modo a reduzir as DAPs [6].

2.2 Eletromiografia e Eletrocardiografia

Nos momentos de atividade, a musculatura recebe um estímulo e responde contraindo suas fibras. Essa resposta gera uma mudança no potencial elétrico em relação a seu estado de repouso e isso pode ser medido e observado eletronicamente. A Eletromiografia (EMG) é uma ferramenta que utiliza esse princípio para monitorar a atividade elétrica de um grupamento muscular e pode ser empregada de diferentes formas, através de técnicas distintas como a Eletromiografia de Superfície (sEMG), na qual este trabalho será focado [8].

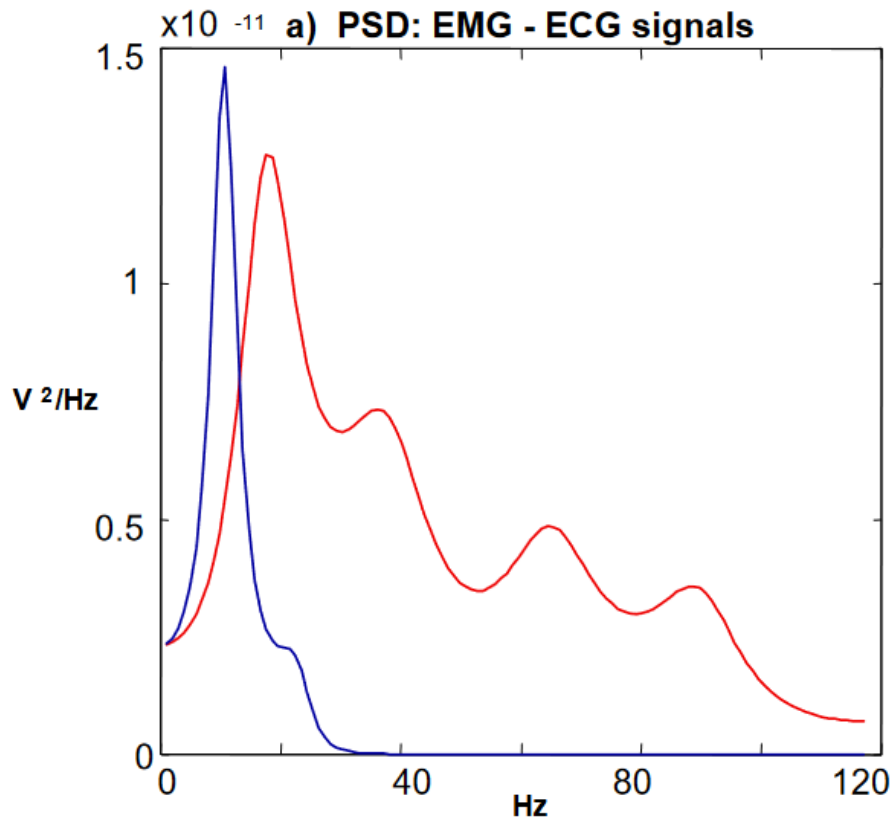
A Eletromiografia de Superfície é uma técnica amplamente reconhecida por ser segura, de fácil aplicabilidade e útil para diversas finalidades. A aquisição da atividade muscular é captada por eletrodos de superfície posicionados na epiderme, acima da musculatura a ser analisada. Contudo, esse sinal apresenta uma amplitude muito baixa e grande interferência de diversas fontes de ruído. Para que a resposta final seja exibida de forma clara e isolando ao máximo a musculatura em foco, o sinal deve passar por etapas de filtragem e amplificação [8].

Uma fonte relevante de artefatos que poluem o sinal EMG é o sinal de eletrocardiograma (ECG). O ECG é resultado de uma ocorrência similar ao eletromiográfico porém, de forma mais específica, representa a ativação da musculatura cardíaca [9].

O espectro de um sinal eletromiográfico fica em torno de $10Hz$ e $500Hz$, tendo uma maior distribuição de energia no intervalo de $20Hz$ a $150Hz$. Já um sinal ECG ocupa uma banda entre $0.1Hz$ e $100Hz$, mais concentrado em $0.1Hz$ a $45Hz$. Além disso, dependendo da região onde são extraídos os sinais, o ECG pode apresentar uma maior amplitude em relação ao EMG [9].

Na Figura 2.2 é possível observar o gráfico da densidade espectral de potência do sinal EMG em vermelho e do sinal ECG em azul.

Figura 2.2: Densidade espectral de potência EMG(Vermelho) - ECG(Azul).



Fonte: Research Gate [10]

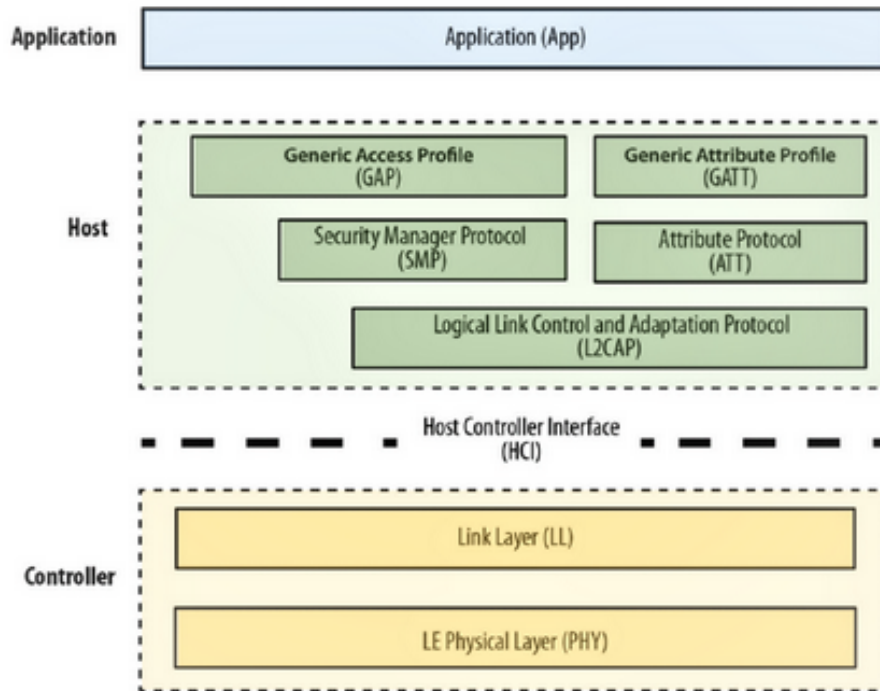
2.3 Bluetooth Low Energy

Bluetooth *Low Energy* (BLE), ou Bluetooth Smart, é uma versão relativamente recente da tecnologia Bluetooth, desenvolvida com foco na diminuição de consumo de energia. Para atingir esse resultado, existem algumas diferenças entre o BLE e sua versão clássica, como a capacidade de transferência de dados. Por conta disso o Bluetooth *Low Energy* é uma opção muito mais voltada para aplicações com uma taxa de transmissão de dados mais moderada e que se beneficie dessa melhor eficiência energética, sendo amplamente utilizada como opção para soluções de Internet das Coisas (*Internet of Things* ou IoT) [11].

A arquitetura do BLE é dividida em Controle, *Host* e aplicação, como pode ser visto na Figura 2.3. Por estarem mais próximas da camada de aplicação, as camadas Perfil de Acesso Genérico ou GAP (Generic Access Profile) e Perfil de

Atributo Genérico ou GATT (Generic Attribute Profile) especificam e gerenciam serviços e atributos. Dessa forma seu entendimento é bastante relevante para a compreensão do funcionamento e da topologia de comunicação [11].

Figura 2.3: BLE pilha de protocolos.



Fonte: Research Gate [12]

O Perfil de Acesso Genérico permite dispositivos BLE operarem uns com os outros, permitindo serem descobertos, fazer *broadcast* de dados e estabelecer conexões seguras de forma padronizada. Existem diferentes aspectos que são definidos pelo GAP que impõem restrições e demandas de funcionamento [13].

A camada GATT é considerada a espinha dorsal da comunicação BLE e é responsável por estruturar a forma como o cliente e o servidor trocam informações. Inicialmente o cliente realiza a descoberta do servidor, para que em posse dos atributos possa fazer leituras e escritas. O servidor recebe requisições, cria usuários e armazena-os, tornando disponíveis para o cliente [13].

A comunicação é baseada em objetos em alto nível, estruturados de forma encapsulada. Cada um desses elementos possui atributos como UUID (identificador

universal único) de formato string de 16, 32 ou 128 *bits* e *Value* que armazena os valores referentes aquele UUID de até 2, 4 ou 16 *bytes* respectivamente [13]. Mais especificamente esses três elementos são:

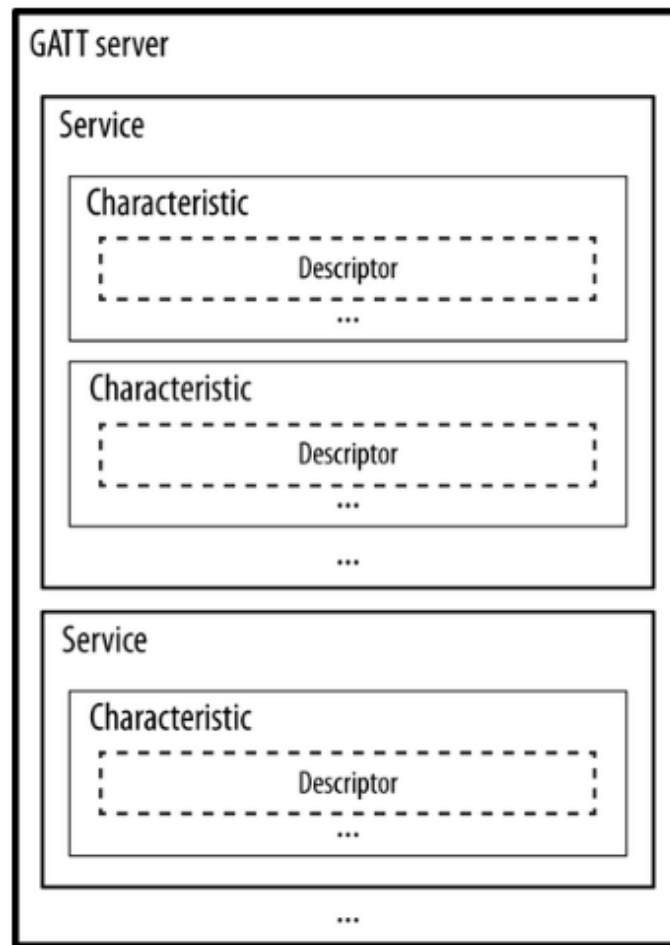
- Serviço;
- Característica;
- Descritor.

Serviços são usados para estruturar dados em entidades lógicas distintas. Eles podem conter nenhuma ou múltiplas características [13].

Encapsulada num serviço, a característica é a entidade de mais baixo nível no GATT e um de seus pontos principais de interação. Ela contém um único ponto de dado (apesar de poder conter um vetor de dados correlacionados), sendo possível escrever e ler valores contidos em seu corpo [13].

Por fim, uma característica pode ou não conter um ou mais descritores que desempenham o papel de guardar e prover ao cliente metadados da característica na qual está contido [13].

Figura 2.4: Hierarquia de encapsulamento do objeto GATT.



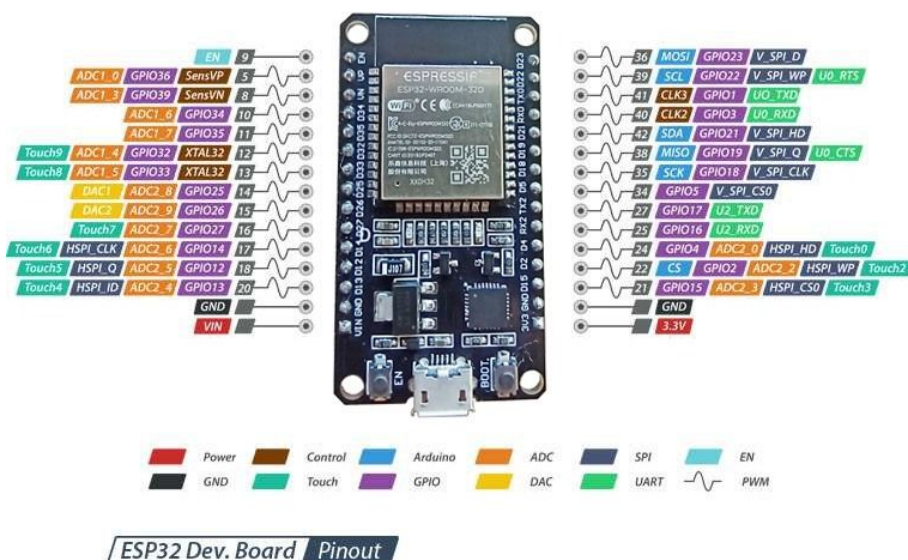
Fonte: O'Reilly [14]

2.4 ESP32 DEVKITV1

A ESP32 Devkit é uma placa de desenvolvimento muito utilizada em soluções de Internet das Coisas (IoT) e conta com um microprocessador de baixa potência com suporte para WiFi e Bluetooth integrados. Além disso, outros periféricos como memória *flash*, ADC, DAC e GPIO a tornam uma solução interessante para o desenvolvimento do trabalho proposto [15].

A programação da ESP32 também pode ser feita de forma bastante simples com suporte ao *framework* do Arduino, utilizando C/C++ e dispõe de ampla quantidade de bibliotecas de integração como do Bluetooth e display LCD [16].

Figura 2.5: ESP32 pinout.



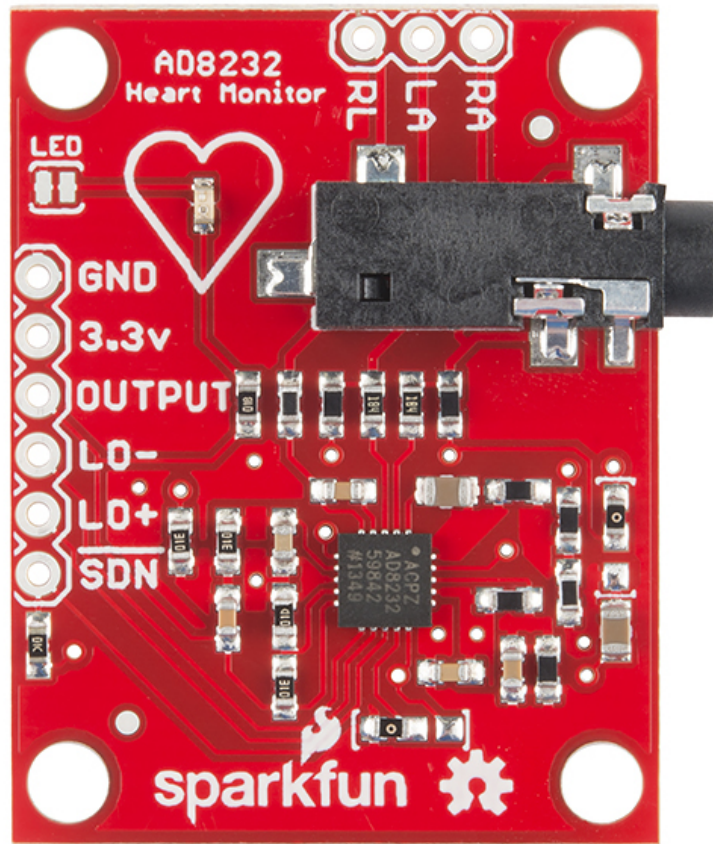
Fonte: Lobo da Robotica [17]

2.5 Módulo ECG

O AD8232 é um condicionador de sinais integrado para ECG e outras medições de biopotencial. Trata-se de um chip bastante popular para essa finalidade, sendo muito usado em módulos monitores cardíacos, que implementam toda circuitaria necessária para utiliza-lo em uma de suas várias possíveis configurações [18].

Como seu foco de uso está em monitorar sinais cardíacos a maioria dos módulos utilizando este chip implementa soluções voltadas para o sinal ECG. Como será mostrado posteriormente, apesar da utilização do módulo AD8232 não ser voltada especificamente para EMG, é possível extrair componentes desse sinal, já que os espectros de ambos ECG e EMG são relativamente próximos e contam com trechos de sobreposição [9].

Figura 2.6: Módulo monitor cardíaco AD8232.

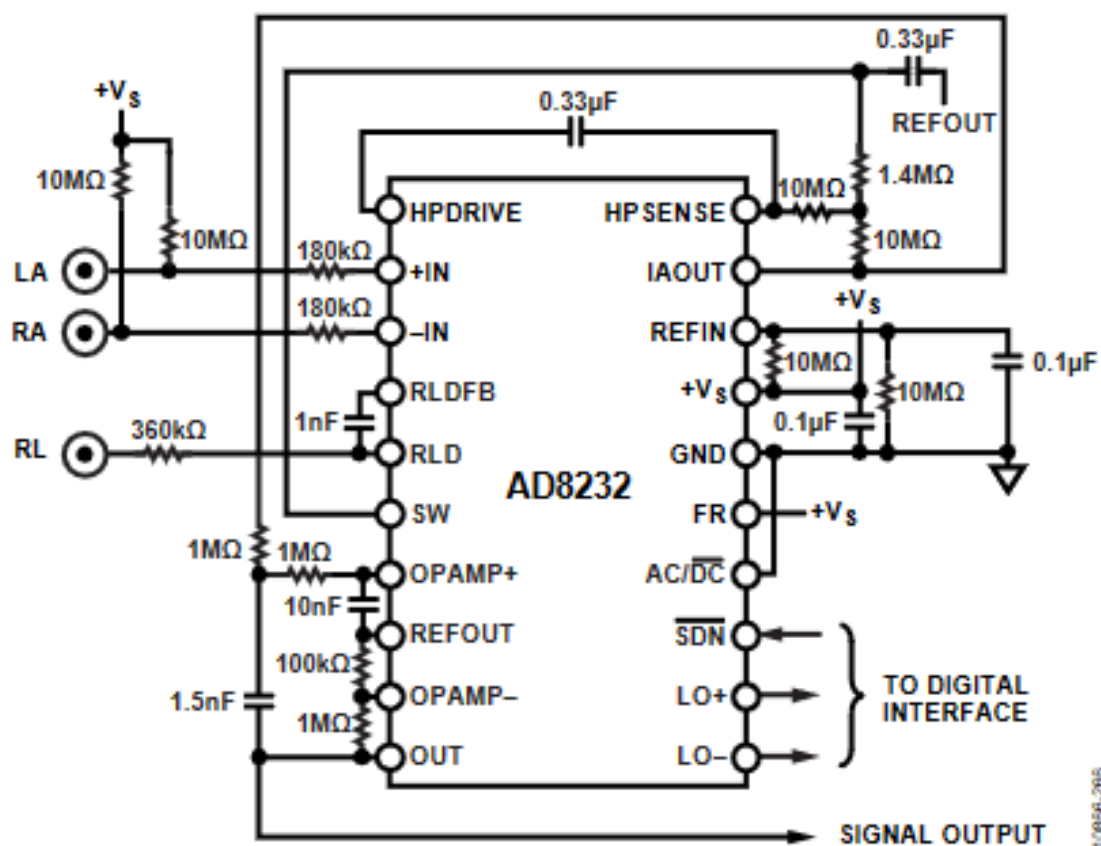


Fonte: Sparkfun [19]

Inicialmente um amplificador de instrumentação devolve a diferença entre os sinais dos eletrodos posicionados na musculatura, vistos na Figura 2.7 como LA e RA nos terminais $+IN$ e $-IN$ respectivamente. Ele garante não só a eliminação do *offset* dos eletrodos como também amplifica essa diferença de sinais [18].

A configuração do circuito do módulo usado na aplicação é a de monitor cardíaco. Uma etapa de filtragem com um filtro passa faixa para isolar ao máximo a banda de frequência dos sinais cardíacos (cujo espectro se assemelha aos miográficos) e amplificação para deixar o sinal numa amplitude adequada para ser lido e processado devidamente [18].

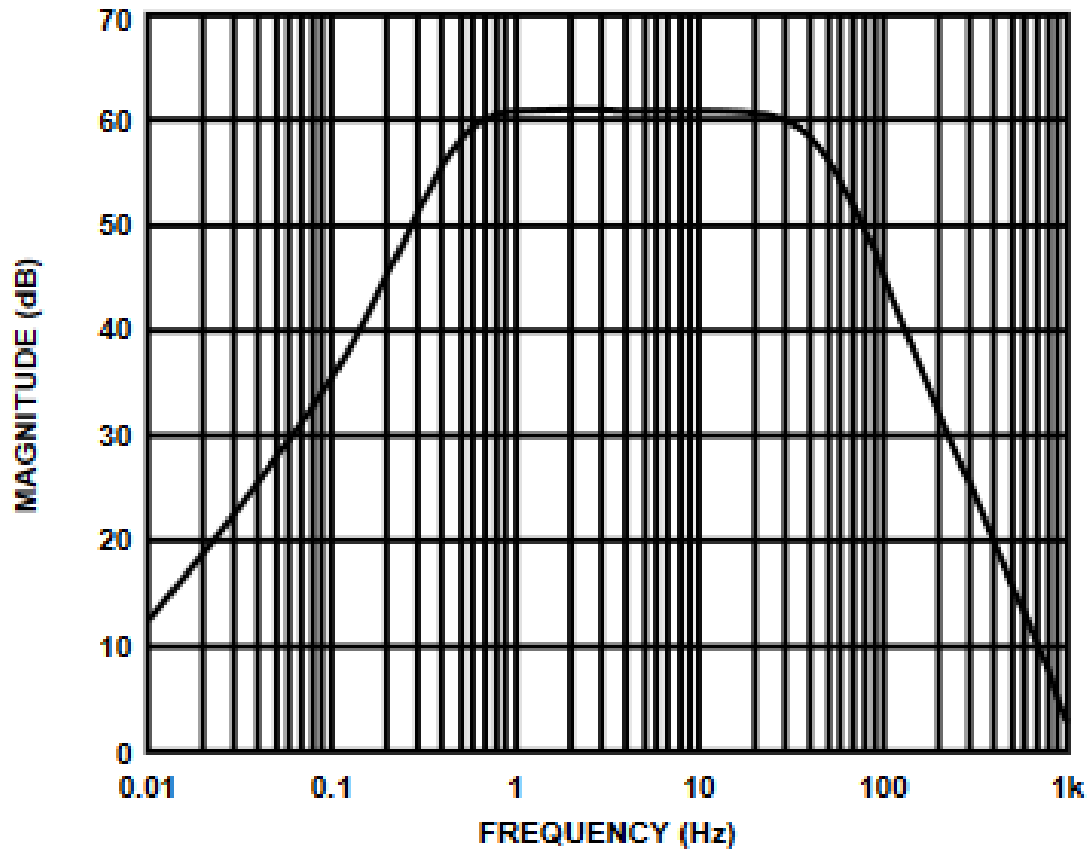
Figura 2.7: Circuito ECG AD8232.



Fonte: Analog [20]

O filtro passa-alta de segunda ordem tem frequência de corte de $0.5Hz$ para bloquear a componente CC e o passa baixas também de segunda ordem com a frequência de corte de $40Hz$, de maneira que a magnitude da resposta em frequência pode ser observada na Figura 2.8 [18].

Figura 2.8: Gráfico da magnitude da resposta em frequência do módulo monitor cardíaco.



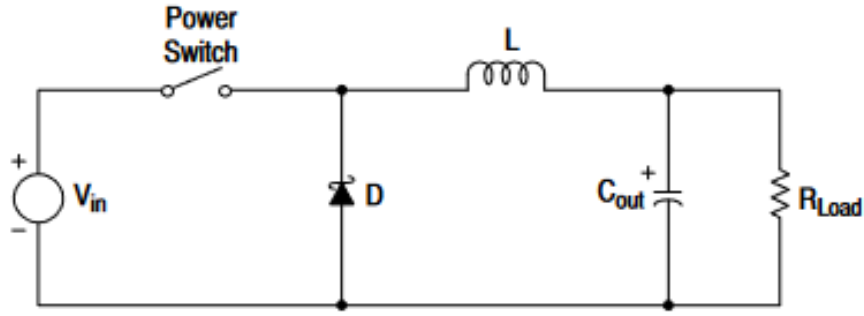
Fonte: Analog [21]

2.6 Conversor *Buck*

Step-Down ou *Buck* é um circuito conversor de tensão de corrente contínua que diminui a tensão de entrada, funcionando como uma fonte chaveada [22].

Na Figura 2.9 é possível observar um *Step-Down* básico. A seguir é feita uma breve análise de funcionamento desse circuito básico para tornar mais claro o entendimento e a relação com a solução de fato utilizada neste projeto.

Figura 2.9: Circuito Buck.



Fonte: Onsemi [23]

O circuito pode ser analisado em dois períodos distintos de tempo. Com a chave fechada, a tensão constante de entrada está conectada com o restante do circuito. Dessa forma a corrente no indutor cresce linearmente e energia magnética é armazenada pelo indutor [24].

No período seguinte a chave abre, invertendo a polaridade no indutor, fazendo com que o diodo seja polarizado diretamente e garantindo que o loop de corrente no resistor de carga seja mantido [24].

A tensão de saída é regulada a partir do ciclo de trabalho controlado pelo chaveamento. Sendo V_{out} a tensão de saída, V_{in} a tensão de entrada e D o ciclo de trabalho (*duty cycle*) controlado pelo chaveamento, de forma simplificada, a relação da tensão de saída é determinada aproximadamente pela Equação 2.1. Esta aproximação fica cada vez melhor a medida que a frequência de chaveamento aumenta. Contudo, geralmente um MOSFET, tem uma frequência máxima de operação [24].

$$V_{out} = DV_{in} \quad (2.1)$$

Um circuito comercialmente popular utilizado para esta função é o módulo regulador de tensão LM2596. É capaz de receber 3,2V a 40V na entrada, devolvendo uma tensão de saída de 1,5V a 35V.

Sendo um conversor chaveador, apresenta uma vantagem quanto a eficiência energética em relação a outros reguladores de tensão lineares, permitindo assim

um maior aproveitamento do tempo de uso das baterias, menor aquecimento do dispositivo além de outras vantagens decorrentes dessa maior eficiência [22].

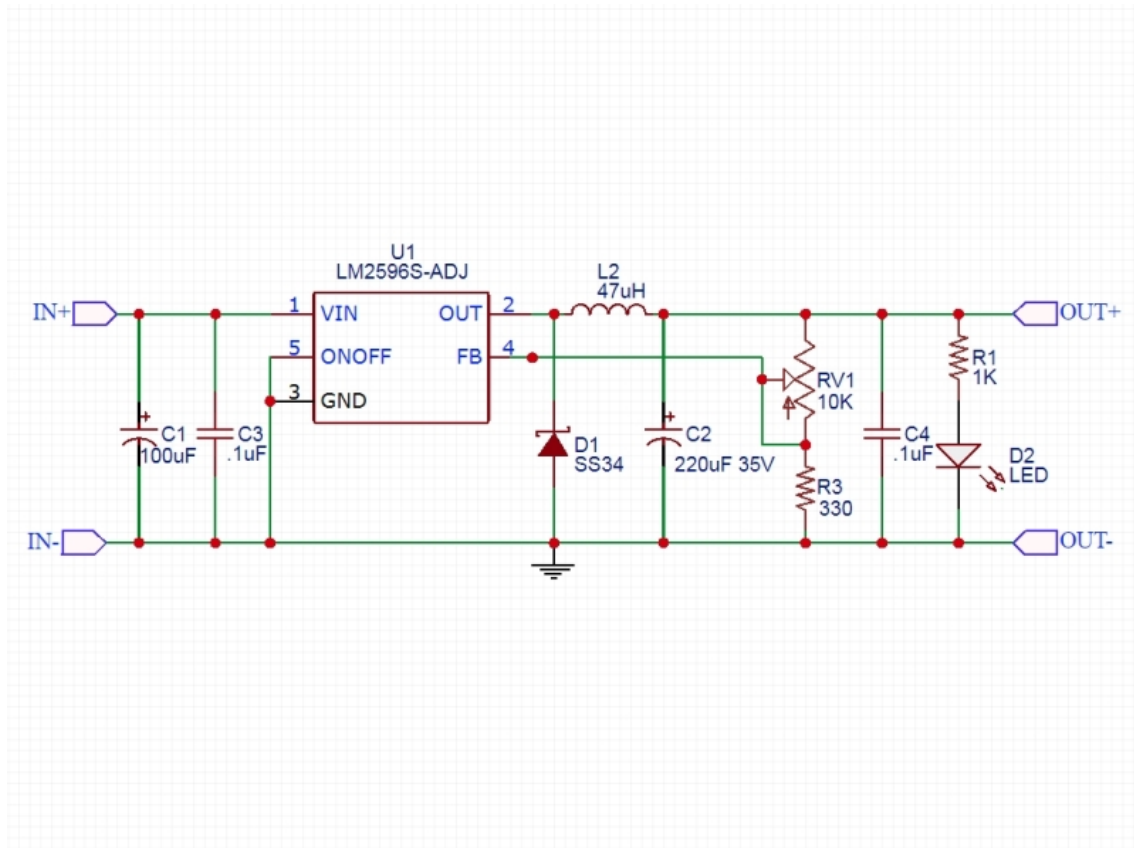
Figura 2.10: Módulo Step Down LM2596.



Fonte: Easytronics [25]

O esquemático do módulo LM2596 pode ser observado na Figura 2.11.

Figura 2.11: Módulo DC-DC LM2596.



Fonte: Elaborada pelo autor.

2.7 Python

A linguagem escolhida para escrita do *software* foi o Python por motivos tais como a grande disponibilidade de documentações, bibliotecas, ferramentas de apoio e informações de forma geral.

É relevante citar o papel de algumas ferramentas e bibliotecas que foram importantes para o desenvolvimento do programa final.

2.7.1 Bleak

A biblioteca Bleak comporta elementos que ajudam a implementar as funcionalidades BLE. Criou-se uma classe Connection que encapsula a BleakClient para realizar o gerenciamento da comunicação e leitura de dados [26].

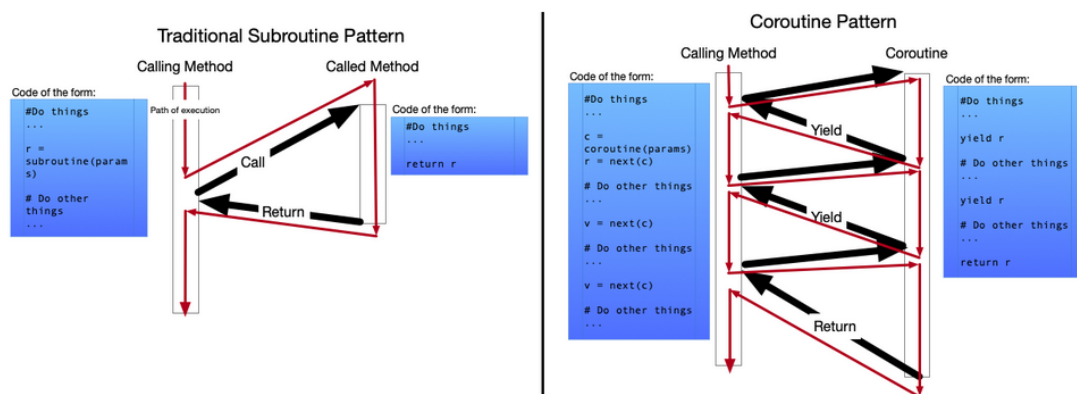
2.7.2 Matplotlib

Matplotlib é bastante popular e reconhecida na literatura e documentação científica como uma excelente biblioteca para auxílio visual, sendo capaz de criar ilustrações personalizadas para visualizar diferentes tipos e representações de dados. Neste projeto ela foi utilizada para plotar o gráfico em tempo real dos valores eletromiográficos ou os gráficos do banco de arquivos de formato *comma-separated values* (CSV) com o histórico de informações [27].

2.7.3 Asyncio

Para contornar o bloqueio da execução do código devido a eventos de entrada e saída (IO), optou-se pelo uso do Asyncio como forma de implementar *tasks* assíncronas para o *plot* dos gráficos em tempo real e gerenciamento da comunicação entre cliente e servidor. Dessa forma, uma tarefa como a exibição do gráfico não mais bloqueia a execução das demais funções [28].

Figura 2.12: Demonstração de fluxo de código usual vs assíncrono.



Fonte: Github [29]

2.7.4 Pandas

Outra ferramenta *open – source* e bastante relevante é a Pandas, muito utilizada para manipulação e análise de dados. Neste trabalho a Pandas será utilizada para

formatar os dados EMG das medições em estruturas Dataframe que posteriormente serão convertidas e gravadas em arquivos CSV [30].

A escolha de utilizar um banco local de arquivos *CSV* se deu pela facilidade de identificar, selecionar e tratá-los de forma simples, tanto via código como através de ferramentas de leitura de planilhas. A pasta *track_record* armazena os documentos que formam o histórico de atividades.

Capítulo 3

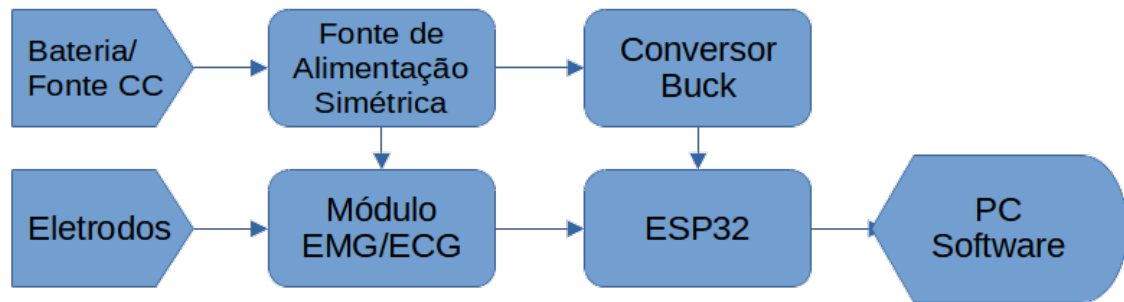
Desenvolvimento

A PROPOSTA deste capítulo é apresentar uma visão geral do sistema desenvolvido, explicitando a relação e integração entre os componentes de *hardware* e *software* utilizados.

3.1 Sistema Proposto

A ideia geral proposta para auxiliar na solução do problema foi inicialmente simplificada de acordo com o diagrama da Figura 3.1, onde de forma generalizada, são divididos em aspectos funcionais os blocos no qual o sistema como um todo é constituído.

Figura 3.1: Diagrama geral.

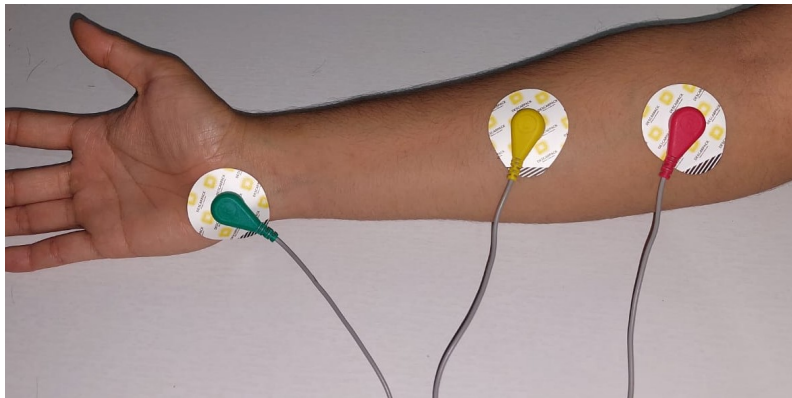


Fonte: Elaborada pelo autor.

3.1.1 Aquisição do Sinal

Para a extração do sinal de entrada, foram postos três eletrodos em pontos da musculatura do antebraço como na Figura 3.2. Esse músculo foi escolhido por ser um grupamento relativamente grande, fácil de posicionar os eletrodos e capazes de gerar um sinal de maior intensidade, facilitando também a distinção da leitura das contrações em meio ao ruído.

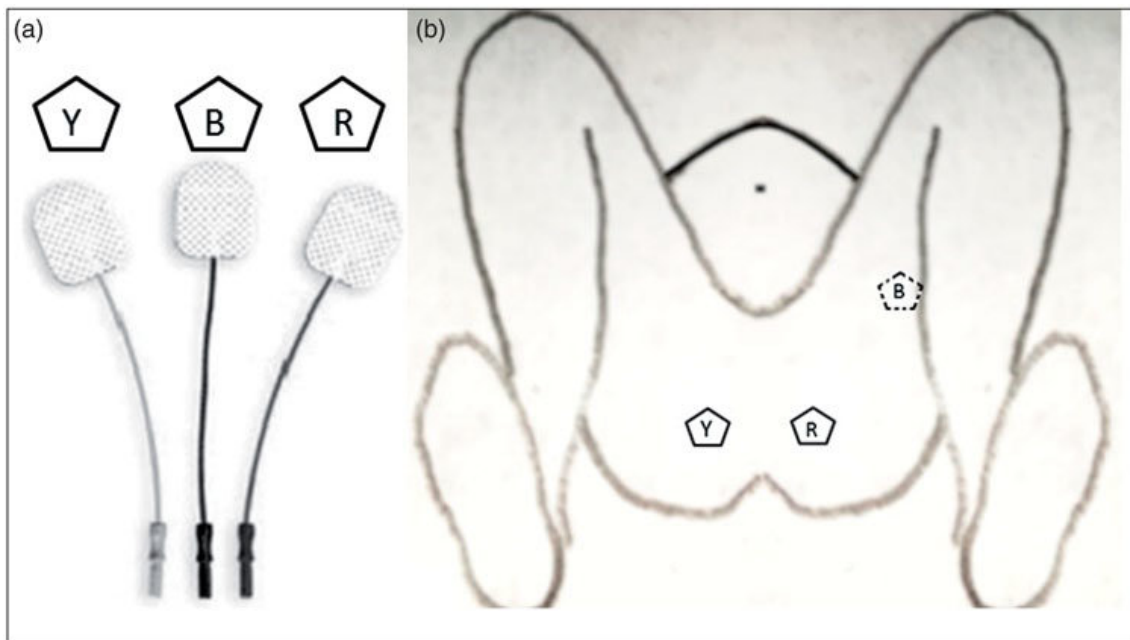
Figura 3.2: Eletrodos posicionados sobre musculatura do antebraço.



Fonte: elaborada pelo autor.

Todavia, na aplicação prática do Biofeedback, os eletrodos seriam posicionados na região dos músculos apropriados, tendo a disposição mostrada na Figura 3.3.

Figura 3.3: Posicionamento correto dos eletrodos para uma aplicação de Biofeedback da região pélvica.

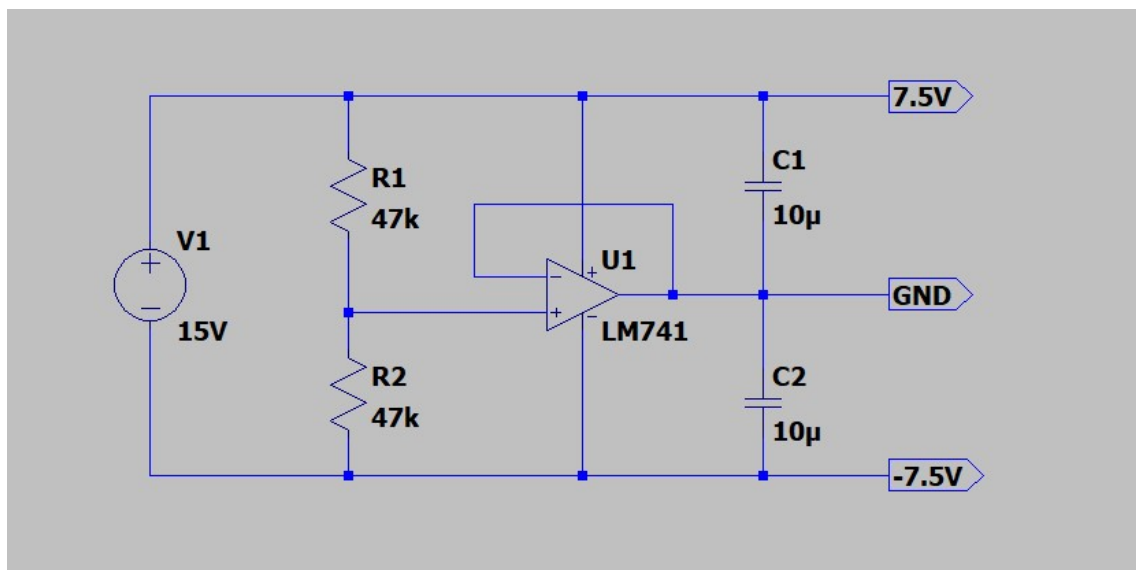


Fonte: Research Gate [31]

3.1.2 Circuito de Alimentação

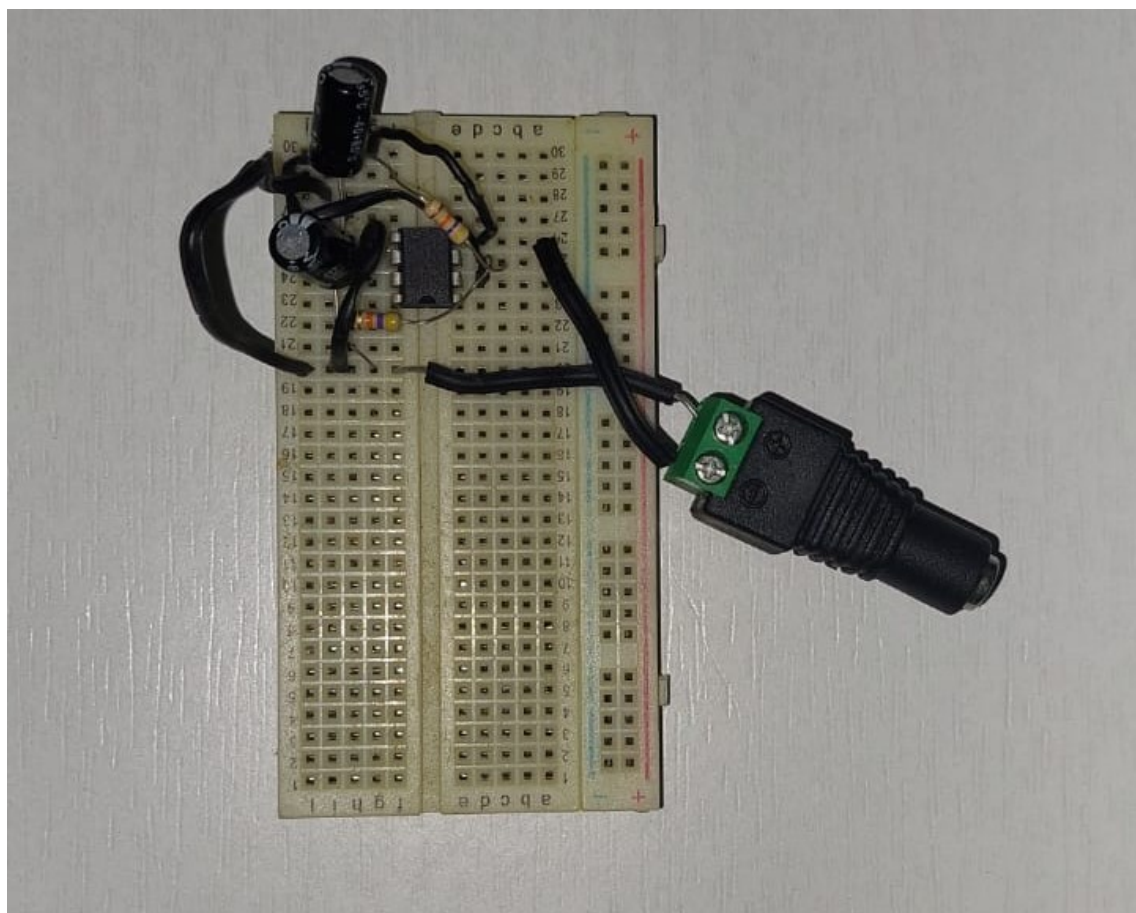
Na unidade de alimentação foi implementado, numa protoboard, um circuito com papel de fonte CC simétrica. Foi utilizada uma fonte de 15V CC como entrada, provendo na saída $+7,5V$, $-7,5V$ e o GND para o módulo de processamento do sinal.

Figura 3.4: Esquemático do circuito de alimentação simétrica.



Fonte: Elaborada pelo autor.

Figura 3.5: Circuito de alimentação simétrica na protoboard.



Fonte: Elaborada pelo autor.

De forma complementar, foi usado o módulo conversor LM2596 para readequar a tensão positiva provida pela fonte simétrica para alimentar o NodeMCU em seu pino de 5V.

É relevante comentar que a fonte de alimentação de 15V CC foi utilizada por motivos de disponibilidade no momento da confecção do projeto. Alternativamente poderia ter sido usada uma entrada de aproximadamente 10V CC, resultando em 5V CC na saída da fonte de alimentação simétrica, tornando desnecessário o uso do módulo *Buck*.

3.1.3 Processamento do Sinal

Recebendo os sinais captados pelos eletrodos, a etapa de filtragem e amplificação condiciona o sinal de entrada e explicita a banda de interesse para um sinal muscular.

Nesse estudo foram comparados os resultados com a utilização de um módulo EMG e um ECG.

O circuito EMG desenvolvido pode ser analisado em 3 etapas ou blocos conectados em cascata. São eles, da esquerda para a direita respectivamente:

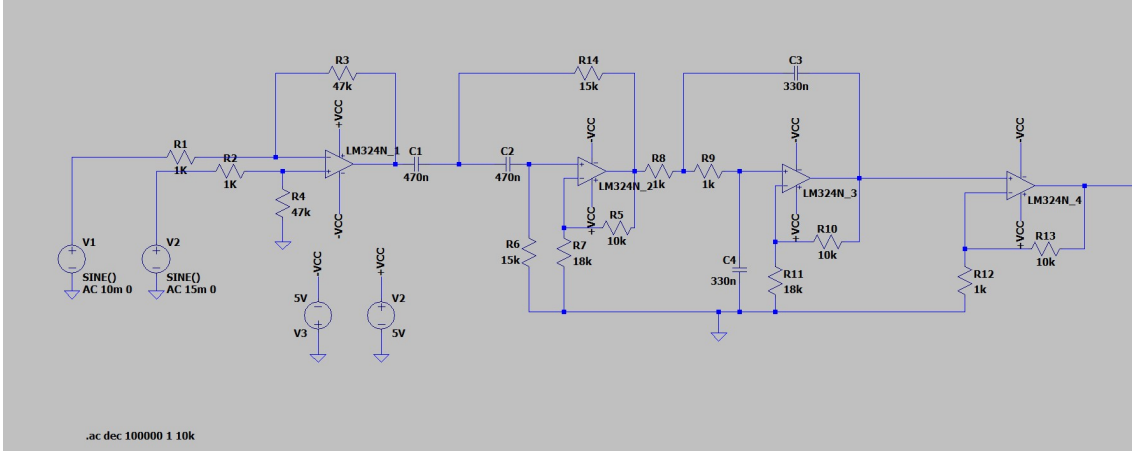
- Amplificador diferencial (LM324N_1);
- Filtro passa-faixa de segunda ordem de frequências de corte $F_{C1} = 21Hz$ e $F_{C2} = 472Hz$ (LM324N_2 e LM324N_3);
- Amplificador não-inversor com ganho ajustável (LM324N_4).

Esta organização é vista na representação da Figura 3.6.

O primeiro AMP-OP (LM324N_1) é um amplificador na configuração diferencial clássica com quatro resistores [32], cuja função de transferência pode ser expressa por:

$$V_{out} = \frac{R_4}{R_2 + R_4} \times \frac{R_1 + R_3}{R_1} \times V_2 - \frac{R_3}{R_1} V_1, \quad (3.1)$$

Figura 3.6: Esquemático do circuito no LTSpice.



Fonte: Elaborada pelo autor.

com V_{out} sendo a saída do AMP-OP e na condição de $R_1 = R_2$ e $R_3 = R_4$, é possível simplificar a equação, resultando em:

$$V_{out} = \frac{R_3}{R_1}(V_2 - V_1). \quad (3.2)$$

Na implementação prática os resistores estão muito próximos de respeitar a condição acima definida, de forma que é possível assumi-la verdadeira para o contexto do projeto.

A equação 3.2 deixa explícito que a saída do amplificador será de fato a diferença dos sinais de entrada amplificados por um fator multiplicativo.

O segundo e terceiro AMP-OPs (LM324N_2 e LM324N_3) compõem o filtro passa-banda (ou passa-faixa) de segunda ordem, formado por um passa-alta cascadeado com um passa-baixa respectivamente [33]. A banda passante é definida pelas frequências de corte de ambos filtros, que de forma simplificada e para resistores e capacitores de mesmo valor, é expressa por:

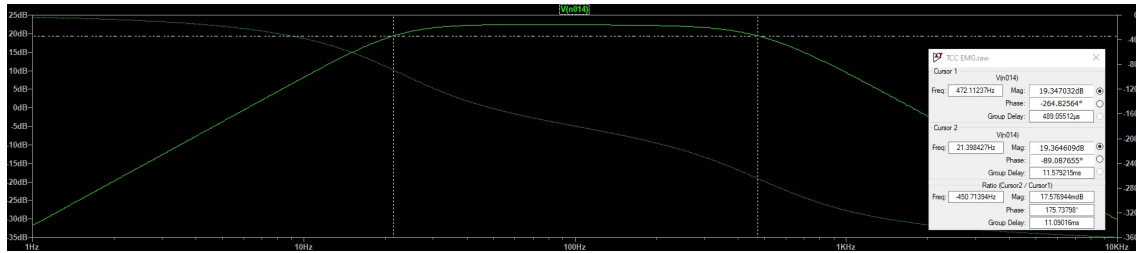
$$F_C = \frac{1}{2\pi RC}. \quad (3.3)$$

Dessa maneira, aplicando a equação 3.3 para o circuito simulado a frequência de corte para o PA é $F_{C1} \approx 22,58Hz$ e para o PB é $F_{C2} \approx 482,29Hz$.

Por último, o LM324N_4 está na configuração de amplificador não-inversor. O resistor de $10K\Omega$ utilizado na simulação foi substituído por um potenciômetro de $100K\Omega$ em série com um resistor de $47K\Omega$ para permitir um ganho ajustável.

O resultado obtido da análise AC é apresentado na Figura 4.1. Essa resposta valida o comportamento em frequência esperado do sistema com uma banda passante entre $21Hz$ e $472Hz$ aproximadamente [9].

Figura 3.7: Resultado de análise AC realizada no LTSpice.



Fonte: Elaborada pelo autor.

3.1.4 Firmware

Após a etapa de processamento, a ESP32 recebe o sinal analógico em um de seus pinos GPIO, que é então, convertido para sinal digital pelo ADC da ESP.

O *firmware* implementado configura a ESP32 como um servidor BLE, gerenciando o anúncio e a notificação dos valores do ADC a serem recebidos pelo computador. enquanto cliente Bluetooth. No desenvolvimento, utilizou-se a ferramenta PlatformIO integrada ao ambiente do Visual Studio Code, que serve como uma plataforma para facilitar o desenvolvimento de *software* embarcado. Neste trabalho o PlatformIO cumpriu o papel de integrar ferramentas, bibliotecas e *framework* em um único ambiente, permitindo utilizar o *framework* arduino com o conjunto de bibliotecas para Bluetooth para gerar o *firmware* final a ser gravado na memória da placa de desenvolvimento ESP32 Devkit.

As classes *MyServerCallbacks* e *MyCallbacks*, descrevem os métodos de resposta para os eventos das instâncias do servidor e característica, respectivamente.

Ambos métodos da primeira classe apenas atribuem o valor a uma variável global booleana para indicar quando houve conexão (*onConnect*) ou desconexão (*onDisconnect*), que será conferida no loop principal.

Figura 3.8: Classes com métodos de callback.

```
21
22 class MyServerCallbacks: public BLEServerCallbacks {
23     void onConnect(BLEServer* pServer) {
24         deviceConnected = true;
25     };
26
27     void onDisconnect(BLEServer* pServer) {
28         deviceConnected = false;
29     }
30 };
31
32 class MyCallbacks: public BLECharacteristicCallbacks {
33     void onWrite(BLECharacteristic *pCharacteristic) {
34         std::string rxValue = pCharacteristic->getValue();
35
36         if (rxValue.length() > 0) {
37             Serial.println("*****");
38             Serial.print("Received Value: ");
39             for (int i = 0; i < rxValue.length(); i++)
40                 Serial.print(rxValue[i]);
41
42             Serial.println();
43             Serial.println("*****");
44         }
45     }
46 };
47
```

Fonte: Elaborada pelo autor.

Na rotina *setup* instancia-se um objeto da classe *BLEDevice*, que inicializa um dispositivo Bluetooth *Low Energy* com o nome passado por parâmetro. O método *createServer* retorna uma nova instância do servidor Bluetooth e habilita seus eventos de *callback* como conexão ou desconexão, através do *setCallbacks*. Associado a este servidor é iniciado um serviço, que por sua vez, contém uma característica com a propriedade *notify*, exposta pelo servidor Bluetooth. Com o servidor e todo seu conteúdo devidamente configurados, ele começa a anunciar (*advertise*) e passa a ficar visível para que o cliente possa se conectar.

Figura 3.9: Trecho do código com função setup.

```
49 void setup() {  
50     Serial.begin(115200);  
51  
52     BLEDevice::init("EMG");  
53  
54     pServer = BLEDevice::createServer();  
55     pServer->setCallbacks(new MyServerCallbacks());  
56  
57     BLEService *pService = pServer->createService(SERVICE_UUID);  
58  
59     pTxCharacteristic = pService->createCharacteristic(  
60         CHARACTERISTIC_UUID_TX,  
61         BLECharacteristic::PROPERTY_NOTIFY  
62     );  
63  
64     pTxCharacteristic->addDescriptor(new BLE2902());  
65  
66     BLECharacteristic * pRxCharacteristic = pService->createCharacteristic(  
67         CHARACTERISTIC_UUID_RX,  
68         BLECharacteristic::PROPERTY_WRITE |  
69         BLECharacteristic::PROPERTY_NOTIFY  
70     );  
71  
72     pRxCharacteristic->setCallbacks(new MyCallbacks());  
73  
74     pService->start();  
75  
76     pServer->getAdvertising()->start();  
77     Serial.println("Waiting a client connection to notify...");  
78 }  
79
```

Fonte: Elaborada pelo autor.

No *loop* principal são checados três casos. O primeiro serve para o momento em que a conexão é estabelecida. Após conectado, é estabelecida uma rotina onde é feita a leitura do valor do ADC naquele instante. Com isso altera-se a característica BLE de acordo com o valor lido e envia-se uma notificação (*notify*). Por último tem o caso que garante a tentativa de reconexão (*startAdvertising*), se um evento de *callback* para desconexão ocorrer.

Figura 3.10: Trecho do código com função loop.

```
80 void loop() {
81     if (deviceConnected)
82     {
83         adcValue = analogRead(15);
84         pTxCharacteristic->setValue(adcValue);
85         pTxCharacteristic->notify();
86         delay(30); // bluetooth stack will go into congestion, if too many packets are sent
87     }
88
89     // disconnecting
90     if (!deviceConnected && oldDeviceConnected)
91     {
92         delay(500); // delay necessary for bluetooth stack to get things ready
93         pServer->startAdvertising(); // restart advertising
94         Serial.println("start advertising.");
95         oldDeviceConnected = deviceConnected;
96     }
97
98     // connecting
99     if (deviceConnected && !oldDeviceConnected)
100     {
101         Serial.print("Connected to client. ");
102         oldDeviceConnected = deviceConnected;
103     }
104 }
105
```

Fonte: Elaborada pelo autor.

3.1.5 Software

Na extremidade final do sistema, o *software* em Python tenta estabelecer uma conexão com a ESP32, recebe os dados da eletromiografia, os exibe de forma gráfica e salva os valores após cada seção.

A classe *Connection* encapsula um objeto *BleakClient* que contém atributos e métodos relativos ao cliente BLE. Ela também será responsável por implementar os métodos que gerenciam a conexão, *callbacks*, desconexões e leitura dos dados provenientes das notificações.

Figura 3.11: Trecho do código fonte da classe *Connection*.

```
29 class Connection:
30
31     client: BleakClient = None
32
33 > def __init__(...
66
67
68 > def on_disconnect(self):...
73
74 > async def cleanup(self):...
79
80 > async def manager(self):...
87
88 > async def connect(self):...
10
11 > async def select_device(self):...
30
31 > def record_info(self):...
37
38 > def clear_lists(self):...
42
43 > def notification_handler(self, sender: str, data: Any):...
62
63
```

Fonte: Elaborada pelo autor.

Há também uma segunda classe *DataToFile*, que lida com o armazenamento dos dados, compondo uma estrutura *dataframe* do pacote Pandas e salvando num arquivo CSV ao final da execução.

Figura 3.12: Trecho do código fonte da classe *DataToFile*.

```
6 class DataToFile:
7
8     header = ["time", "delay(milliseconds)", "data_value"]
9
10    def __init__(self, file_name):
11        self.file_name = file_name
12        self.columns = ['time', 'delay(milliseconds)', 'data_value']
13        self.df = pd.DataFrame({"time":[],
14                                "delay(milliseconds)":[],
15                                "data_value":[]})
16
17    def append_dataframe(self, time, delay, data_value):
18        temp_df = pd.DataFrame({"time":time,
19                                "delay(milliseconds)":delay,
20                                "data_value":data_value})
21        self.df = self.df.append(temp_df, ignore_index=True)
22
23
24    def write_to_csv(self, file_name):
25        self.df.to_csv(file_name)
26        print(f"Data plotted saved on csv file named: {file_name} .")
27
28
29    def plot_from_csv(self):
30        dt_frame = pd.read_csv(self.file_name)
31        dt_frame.plot()
32        plt.show()
```

Fonte: Elaborada pelo autor.

O fluxo principal de execução está na função *Main*. A interface de usuário foi implementada de forma simples com opções exibidas no console. Dessa forma, inicialmente o programa permite a escolha entre a exibição do gráfico de algum arquivo previamente salvo ou a leitura e apresentação dos dados lidos em tempo real.

Figura 3.13: Trecho do código fonte da função *Main*.

```
165 #####
166 # App Main
167 #####
168
169 def main():
170     #simple script to test both features
171
172     print("Choose option:")
173     print("1 - Plot from csv")
174     print("2 - Read live EMG")
175     print("Option: ")
176     menu_option = int(input())
177
178     if (menu_option == 1):
179         print("Plotting from csv file.")
180         file_list = os.listdir(dir)
181         print('Select file to plot:')
182         [print(str(i) + ' - ' + file) for i, file in enumerate(file_list)]
183         data_to_file = DataToFile(dir + file_list[int(input())])
184         try:
185             data_to_file.plot_from_csv()
186         except TypeError:
187             print("No numeric data to plot")
188     elif (menu_option == 2):
189         # Create the event loop.
190         print("Read live EMG from BLE server.")
191         loop = asyncio.get_event_loop()
192
193         file_name = "{}EMG plot {:Y-%m-%d %Hh%Mm}.csv".format(output_directory, datetime.now())
194         data_to_file = DataToFile(file_name)
```

Fonte: Elaborada pelo autor.

Capítulo 4

Resultados

4.1 Hardware

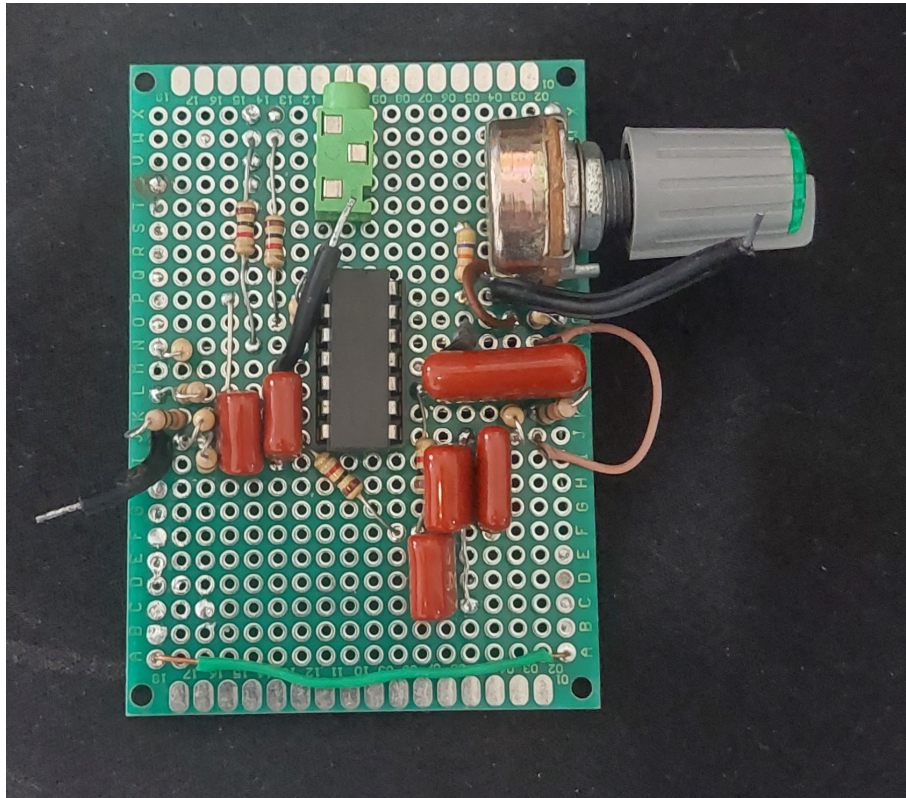
A ideia do circuito EMG, observada na seção 3.1.3, foi implementada em uma PCB de ilhas genérica, tentando-se aproximar ao máximo dos valores usados no circuito projetado no LTSpice. Os seguintes componentes foram utilizados na montagem:

- 1 LM324N;
- 1 Potenciômetro de $100K\Omega$;
- 1 Conector jack P2;
- 5 Resistores de $1K\Omega$;
- 3 Resistores de $47K\Omega$;
- 2 Resistores de $18K\Omega$;
- 2 Resistores de $15K\Omega$;
- 2 Resistores de $10K\Omega$;
- 4 Capacitores de $470nF$;
- 1 Capacitores de $330nF$;

- 1 Capacitores de $100nF$.

O resultado obtido é observado na Figura 4.1.

Figura 4.1: Protótipo do circuito em PCB.



Fonte: Elaborada pelo autor.

Alternativamente ao circuito de processamento desenvolvido e descrito, foi usado durante os testes o módulo ECG com o AD8232, pela facilidade de aplicação e para comparação dos resultados, já que a resposta em frequência deste módulo, em parte, se sobrepõe a faixa de interesse.

4.2 Envio e Armazenamento dos Dados

Os eventos de notificação do servidor BLE foram configurados para 30 ms. A partir dos testes conduzidos, observou que esse foi o menor valor no qual o Bluetooth conseguiu operar, atualizando o valor da característica e liberando a notificação de forma estável. Para valores abaixo dos 30 ms de *delay* no envio, algumas das leituras

realizadas pelo cliente BLE não refletiam os valores corretos, lidos pelo ADC da ESP32.

Para evitar o fenômeno de *aliasing* em componentes de até $450Hz$ do sinal, é necessário uma taxa de amostragem de pelo menos $900Hz$. Como foi possível executar a uma taxa de 30 transmissões por segundo com o BLE, foi preciso notificar 30 valores lidos pelo ADC para cada uma delas [34].

Os atrasos entre os valores lidos pelo *software* foram calculados a partir da diferença entre o momento que o Bluetooth realiza a leitura do valor atual (capturado pela função *datetime.now()*) e o momento da leitura anterior, em mili segundos. Esses atrasos foram armazenados na coluna *delay* do arquivo CSV, gerado ao final da execução do programa, e visto na Figura 4.2. Cada valor da coluna D (*data_value*) corresponde apenas ao primeiro valor lido pelo ADC da ESP32, dentre os 30 valores totais para cada transmissão Bluetooth, como mencionado anteriormente.

Figura 4.2: Exemplo de arquivo CSV gerado pelo software ao final de uma seção.

	A	B	C	D
1		time	delay(miliseconds)	data_value
2	0	2022-03-20 16:43:22.216567	497.976	2130.0
3	1	2022-03-20 16:43:22.293582	77.015	1750.0
4	2	2022-03-20 16:43:22.352594	59.012	1427.0
5	3	2022-03-20 16:43:22.384611	32.017	1372.0
6	4	2022-03-20 16:43:22.417618	33.007	1819.0
7	5	2022-03-20 16:43:22.450626	33.008	2427.0
8	6	2022-03-20 16:43:22.483625	32.999	2675.0
9	7	2022-03-20 16:43:22.516633	33.008	2229.0
10	8	2022-03-20 16:43:22.547640	31.007	1751.0
11	9	2022-03-20 16:43:22.579645	32.005	1360.0
12	10	2022-03-20 16:43:22.611652	32.007	1282.0
13	11	2022-03-20 16:43:22.643660	32.008	1799.0
14	12	2022-03-20 16:43:22.676677	33.017	2397.0
15	13	2022-03-20 16:43:22.709685	33.008	2640.0
16	14	2022-03-20 16:43:22.741693	32.008	2255.0
17	15	2022-03-20 16:43:22.774699	33.006	1871.0
18	16	2022-03-20 16:43:22.806706	32.007	1414.0
19	17	2022-03-20 16:43:22.839715	33.009	1338.0
20	18	2022-03-20 16:43:22.872712	32.997	1663.0
21	19	2022-03-20 16:43:22.905729	33.017	2351.0
22	20	2022-03-20 16:43:22.938738	33.009	2591.0
23	21	2022-03-20 16:43:22.973745	35.007	2299.0
24	22	2022-03-20 16:43:23.007753	34.008	1915.0
25	23	2022-03-20 16:43:23.040752	32.999	1463.0
26	24	2022-03-20 16:43:23.073768	33.016	1335.0
27	25	2022-03-20 16:43:23.111768	38.0	1702.0
28	26	2022-03-20 16:43:23.146784	35.016	2320.0
29	27	2022-03-20 16:43:23.179891	33.107	2655.0
30	28	2022-03-20 16:43:23.211790	31.899	2305.0
31	29	2022-03-20 16:43:23.244917	33.127	1860.0
32	30	2022-03-20 16:43:23.277814	32.897	738.0

Fonte: Elaborada pelo autor.

4.3 Análise dos Gráficos

Na Figura 4.3 é possível observar o gráfico resultante da utilização do módulo ECG com o AD8232. A imagem foi gerada pelo módulo *Pyplot*, da biblioteca *Matplotlib*, durante um teste com os eletrodos posicionados no antebraço.

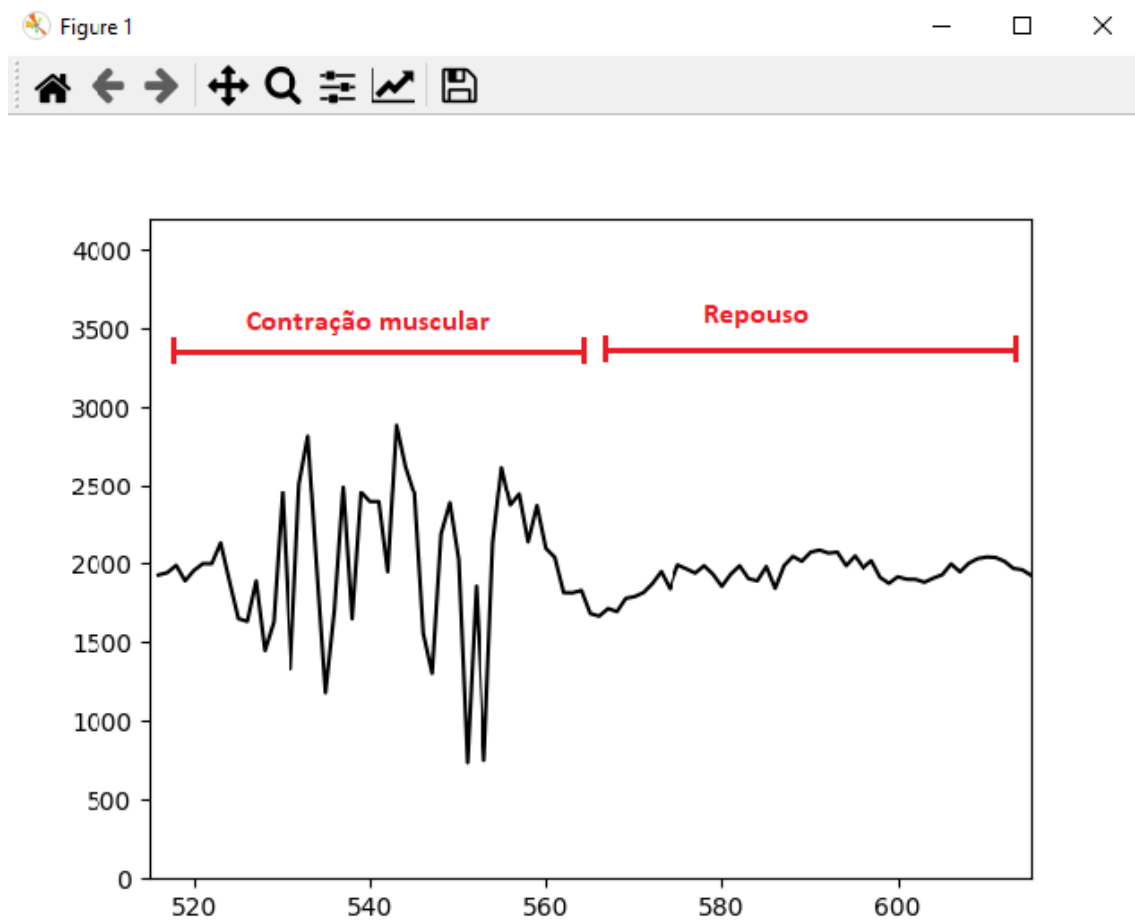
O eixo vertical representando os valores do ADC da ESP32, variando entre 0 (0V) a 4095 (3,3V), já que possui uma resolução de 12 *bits*. Dessa forma, um valor de sinal EMG próximo a 0V (relaxamento total) resulta em 0 no eixo vertical, e um

valor próximo a $3,3V$ (contração máxima) resulta em 4095.

O eixo horizontal está em função do número de amostras recebidas desde o início da execução atual. Para o intervalo visto na Figura 4.3 e considerando um tempo médio de 33 mili segundos entre a leitura de cada amostra, o valor 520 e 600 representam 17,2 segundos e 19,8 segundos após o início da leitura do sinal, respectivamente.

Dessa forma, as variações verticais representam diretamente alterações na intensidade da ativação muscular, enquanto o eixo horizontal atualiza dinamicamente a medida que os valores vão sendo lidos. Vale também ressaltar as duas regiões vistas na Figura 4.3, sendo a região de repouso o momento durante o teste no qual o antebraço estava completamente relaxado e na região de contração a musculatura foi contraída ao máximo.

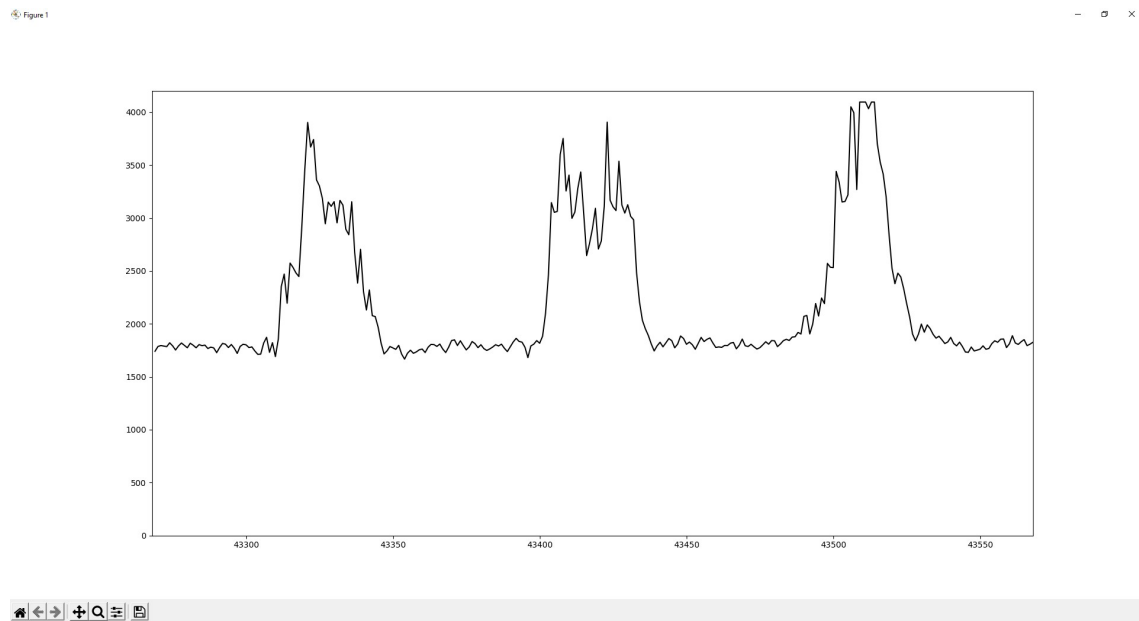
Figura 4.3: Captura de tela demonstrando o resultado observado na aplicação Python, utilizando o módulo ECG AD8232.



Fonte: Elaborada pelo autor.

Já na Figura 4.4 o resultado foi obtido com o uso do circuito EMG implementado na placa padrão, porém de maneira similar ao teste anterior, com o sinal extraído da mesma musculatura no antebraço. Nessa imagem foi capturado um momento do teste onde são executadas três contrações em sequência. Algo que fica claro também é que podemos, de forma muito mais fácil, inferir a intensidade das contrações musculares. Por exemplo, no terceiro pico, os valores são em média muito mais elevados que os outros dois anteriores. Isso faz sentido, uma vez que corresponde a uma contração utilizando mais esforço que as anteriores.

Figura 4.4: Captura de tela demonstrando o resultado observado na aplicação Python, utilizando o circuito EMG desenvolvido.



Fonte: Elaborada pelo autor.

A diferença entre os dois resultados é bastante clara. Como esperado o módulo ECG descreve de maneira pobre a contração muscular, sendo possível identificar quando a mesma ocorre e quando o músculo está em relaxamento. Já para o circuito EMG, o sinal fica mais perceptivelmente dividido entre as regiões de relaxamento e os picos de contração, sendo possível até distinguir entre contrações mais e menos expressivas.

Capítulo 5

Considerações finais

5.1 Conclusão

Dentro do escopo inicial do trabalho, dos objetivos gerais e do que era esperado, o resultado obtido foi um sistema funcional de ponta a ponta, entregando de maneira básica o que foi proposto e uma introdução ao estudo de alguns dos conceitos aplicados e funcionalidades da solução final. Apesar do fechamento da ideia, é notável que diversas melhorias poderiam ser feitas para tornar a solução mais robusta e eficiente na entrega da sua proposta de valor.

5.2 Dificuldades Encontradas

Algumas das principais dificuldades dizem respeito ao acesso limitado a equipamentos para análise dos circuitos e *hardware* do sistema, como osciloscópios e multímetros, dificultando a resolução de certos problemas, de parte da implementação do *hardware* e até da análise dos resultados.

A falta de um equipamento confiável para ter como base na validação entre os resultados tornou impossível uma comparação direta entre a resposta observada em um equipamento de eletromiografia já consolidado no mercado e a solução proposta.

Por parte do *software*, apesar de necessária, a implementação de rotinas assíncronas aumentou o grau de complexidade do que foi desenvolvido.

Parte do desenvolvimento desse trabalho ocorreu durante o período de pandemia da Covid-19, o que resultou em restrições, impactando negativamente a solução final de diferentes formas. Um exemplo desse impacto foi a impossibilidade de visitas a clínicas, para fossem conduzidos testes e feitas comparações entre os resultados do aparelho EMG desenvolvido e de equipamentos profissionais.

5.3 Trabalhos Futuros

Para objetivos mais a longo prazo, busca-se generalizar o emprego da técnica e tecnologia de eletromiografia e *Biofeedback* para usos mais gerais, expandindo o conceito para auxiliar tratamentos de outros grupamentos musculares e outras enfermidades que possam se valer das informações extraídas da atividade muscular.

Com uma base de dados formada e um sistema em tempo real de captura de dados do usuário, é natural a ideia de implementar melhorias utilizando conceitos de aprendizado de máquina, ciência de dados e análise de dados. Diversas ideias plausíveis e podem ser aplicadas nessa área.

- Aperfeiçoar a classificação das atividades em contração e relaxamento, dispondo dos dados obtidos pelo Biofeedback e possivelmente da demografia do usuário;
- estabelecer e avaliar métricas de melhoria dos pacientes ao longo do tempo/seções;
- desenvolver um modelo de predição para escolher o tipo de treino que otimize os resultados para a realidade de cada usuário.

Esses são apenas alguns exemplos que é possível idealizar rapidamente quando pensa-se em soluções aplicando essas áreas, para melhoria do sistema e do tratamento do problema.

Por fim, todas as ideias e conceitos abordados ao longo desse trabalho são passíveis de aplicações mais generalistas. A ideia inicial de uma solução focada na musculatura pélvica e problemas relacionados a essa região, pode muito bem ser

expandida para tratamentos em outros grupamentos musculares e auxílio em outros tratamentos musculares, flexibilizando e ampliando seus cenários de utilização.

Referências Bibliográficas

- [1] ROIG, J. J. “Prevalência de Incontinência Urinária e Fatores Associados em Idosos Institucionalizados”. 2014. Disponível em: <https://repositorio.ufrn.br/bitstream/123456789/17842/1/JavierJR_DISSERT.pdf>. Acesso em: 08/04/2023.
- [2] “Pathway CTS-2000 Pelvic Floor Biofeedback System – Physician Model”. Disponível em: <<https://www.cmtmedical.com/product/pelvic-muscle-rehab/>>. Acesso em: 13/04/2023.
- [3] Disponível em: <<https://fisioterapiacavallazzi.com.br/wpcontent/uploads/2020/09/22222.jpg>>. Acesso em: 23/04/2023.
- [4] “Centro de Incontinência e Doenças do Assolho Pélvico”. Disponível em: <<https://www.einstein.br/estrutura/centro-incontinencia-doencas-assoalho-pelvico/o-que-e-assoalho-pelvico>>. Acesso em: 08/04/2023.
- [5] Disponível em: <<https://loremcorporis.com.br/wp-content/uploads/2021/05/anatomiaassoalho-pelvicoã300x300.png>>. Acesso em: 23/04/2023.
- [6] IBRAHIM, I. K., TAHER, M. M. A. H. E. M., SHAHEEN, E. M., et al. “Efficacy of biofeedback-assisted pelvic floor muscle training in females with pelvic floor dysfunction”. 2015. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S209050681400058X>>. Acesso em: 13/04/2023.
- [7] JÚNIOR, F. “Incontinência urinária é questão de saúde pública”. 2018. Disponível em: <<https://jornal.usp.br/atualidades/incontinencia-urinaria-e-questao-de-saude-publica/>>. Acesso em: 13/04/2023.
- [8] FRANCESCO, F., ALESSANDRO, D. V. “Surface Electromyography: What Limits Its Use in Exercise and Sport Physiology?” 2020.

Disponível em: <<https://www.frontiersin.org/articles/10.3389/fneur.2020.578504>>. Acesso em: 13/04/2023.

- [9] ABBASPOUR, S., FALLAH, A. “A Combination Method for Electrocardiogram Rejection from Surface Electromyogram”. 2014. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3999703/>>. Acesso em: 08/04/2023.
- [10] Disponível em: <https://www.researchgate.net/publication/3958080_Cardiac_interference_in_myographic_signals_from_different_respiratory_muscles_and_levels_of_activity>. Acesso em: 23/04/2023.
- [11] DA SILVA, M. M. B. “Projeto e desenvolvimento de um protocolo de comunicacao para dispositivo vestivel utilizando Bluetooth Low-Energy”. 2018. Disponível em: <<http://dspace.sti.ufcg.edu.br:8080/jspui/bitstream/riufcg/18925/1/MELISSA%20MARIA%20BARBOSA%20DA%20SILVA%20-%20TCC%20ENG.%20EL%C3%89TRICA%202018.pdf>>. Acesso em: 08/04/2023.
- [12] Disponível em: <https://www.researchgate.net/publication/328541826_Assessment_of_Fitness_Tracker_Security_A_Case_of_Study>. Acesso em: 23/04/2023.
- [13] TOWNSEND, K. “Introduction to Bluetooth Low Energy”. 2018. Disponível em: <<https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>>. Acesso em: 13/04/2023.
- [14] Disponível em: <<https://www.oreilly.com/library/view/getting-started-with/9781491900550/ch04.html>>. Acesso em: 23/04/2023.
- [15] OLIVEIRA, E. “Conhecendo o NodeMCU-32S ESP32”. . Disponível em: <<https://blogmasterwalkershop.com.br/embarcados/esp32/conhecendo-o-nodemcu-32s-esp32>>. Acesso em: 09/04/2023.
- [16] OLIVEIRA, E. “NodeMCU-32S ESP32 – Programando com a IDE do Arduino”. . Disponível em: <<https://blogmasterwalkershop.com.br/embarcados/esp32/nodemcu-32s-esp32-programando-com-a-ide-do-arduino>>. Acesso em: 09/04/2023.

- [17] Disponível em: <https://lobodarobotica.com/blog/esp32-pinout/>. Acesso em: 23/04/2023.
- [18] “Single-Lead, Heart Rate Monitor Front End”. Disponível em: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf>. Acesso em: 09/04/2023.
- [19] Disponível em: <https://www.sparkfun.com/products/12650>. Acesso em: 23/04/2023.
- [20] Disponível em: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf>. Acesso em: 23/04/2023.
- [21] Disponível em: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf>. Acesso em: 23/04/2023.
- [22] HART, D. W. “Introduction to power electronics”. 2011.
- [23] . Disponível em: <https://www.onsemi.com/pdf/datasheet/lm2596-d.pdf>. Acesso em: 23/04/2023.
- [24] “3.0 A, Step-Down Switching Regulator”. . Disponível em: <https://www.onsemi.com/pdf/datasheet/lm2596-d.pdf>. Acesso em: 08/04/2023.
- [25] Disponível em: <https://www.easytronics.com.br/lm2596-conversor-dc-dc-step-down>. Acesso em: 23/04/2023.
- [26] “Bleak”. Disponível em: <https://bleak.readthedocs.io/en/latest/>. Acesso em: 13/04/2023.
- [27] “Matplotlib: Visualization with Python”. Disponível em: <https://matplotlib.org/>. Acesso em: 13/04/2023.
- [28] SOLOMON, B. “Async IO in Python: A Complete Walkthrough”. Disponível em: <https://realpython.com/async-io-python/>. Acesso em: 13/04/2023.
- [29] Disponível em: <https://bbc.github.io/cloudfit-public-docs/asyncio/asyncio-part-1.html>. Acesso em: 23/04/2023.
- [30] “pandas”. Disponível em: <https://pandas.pydata.org/>. Acesso em: 13/04/2023.

- [31] Disponível em: <https://www.researchgate.net/publication/323131520_The_behaviour_of_pelvic_floor_muscles_during_uterine_contractions_in_spontaneous_and_oxytocin-induced_labour>. Acesso em: 23/04/2023.
- [32] “The Differential Amplifier”. Disponível em: <https://www.electronics-tutorials.ws/opamp/opamp_5.html>. Acesso em: 13/04/2023.
- [33] “Second Order Filters”. Disponível em: <<https://www.electronics-tutorials.ws/filter/second-order-filters.html>>. Acesso em: 13/04/2023.
- [34] “Acquiring an Analog Signal: Bandwidth, Nyquist Sampling Theorem, and Aliasing”. Disponível em: <<https://www.ni.com/pt-br/shop/data-acquisition/measurement-fundamentals-main-page/analog-fundamentals/acquiring-an-analog-signal--bandwidth--nyquist-sampling-theorem-.html>>. Acesso em: 13/04/2023.