

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

BRUNO MAIOLLI RAZERA

**DESENVOLVIMENTO DE UM SISTEMA ELETROMIOGRÁFICO
SEM FIO**

CAMPO MOURÃO

2021

BRUNO MAIOLLI RAZERA

**DESENVOLVIMENTO DE UM SISTEMA ELETROMIOGRÁFICO
SEM FIO**

Development of a wireless electromyographic system

Trabalho de conclusão de curso apresentado como requisito para obtenção do título de bacharel em engenharia eletrônica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Me. Lucas Ricken Garcia

Coorientador: Prof. Dr. Reginaldo Ré

CAMPO MOURÃO

2021

BRUNO MAIOLLI RAZERA

DESENVOLVIMENTO DE UM SISTEMA ELETROMIOGRÁFICO SEM FIO

Trabalho de conclusão de curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 10/maio/2021

Marcio Rodrigues da Cunha
Doutorado em Engenharia Elétrica
Universidade Tecnológica Federal do Paraná

André Luiz Regis Monteiro
Doutorado em Engenharia Elétrica
Universidade Tecnológica Federal do Paraná

Lucas Ricken Garcia
Mestrado em Engenharia Elétrica e Informática Industrial
Universidade Tecnológica Federal do Paraná

CAMPO MOURÃO

2021

Dedico esta a todos os meus familiares, amigos e professores, grandes companheiros de jornada, que sempre me deram um excepcional apoio e incentivo durante este ciclo.

AGRADECIMENTOS

Agradeço Deus pela vida e todas as graças concedidas ao longo dessa vida.

Agradeço minha família por todo apoio e suporte para me manter no caminho e trilhar essa jornada magnífica.

Agradeço a todos amigos e amigas que me deram forças, auxílio e me ajudaram a perseverar para nunca desistir.

Agradeço a todos os professores que ajudaram a construir quem sou hoje, por toda experiência e conhecimento transmitidos e por terem se tornado grandes amigos.

Agradeço a UTFPR por todo suporte e abrigo ao decorrer desses 10 anos (ensino técnico e faculdade) que foi minha segunda casa.

RESUMO

A eletromiografia (EMG) é uma importante área de estudos de fisiologia e saúde humana, englobando a corrente elétrica gerada em todo o corpo humano, desde a membrana celular até os grupos musculares. Uma dada corrente elétrica é desencadeada em membranas excitáveis das células musculares, o que gera um potencial de ação. Um conjunto desses potenciais, provindos de células que constituem uma mesma unidade motora, resultam em uma contração muscular. O eletromiógrafo é um aparelho cuja função é a de captar o potencial elétrico de um grupo muscular, permitindo monitoramento, estudo e análise de tal grupo. Os equipamentos atuais possuem em sua maioria fios que são interligados ao corpo humano, atrapalhando a realização de movimentos e ainda sendo fonte de artefatos que geram ruídos que interferem no sinal EMG. Dessa forma, o desenvolvimento de um sistema de eletromiografia capaz de realizar a aquisição, transmissão e recepção sem fio dos estímulos elétricos, apresentando-os em uma interface de interação homem-máquina, tornando interessante e possivelmente viável para o mercado atual. Para a construção desse sistema foi desenvolvido um sensor eletromiográfico com um amplificador de instrumentação, filtros ativos, circuito condicionador de sinal e um microcontrolador ESP12F, responsável aquisição e transmissão Wi-Fi por meio de protocolo UDP do sinal. A interface gráfica foi desenvolvida em Python, utilizando os recursos PyQT5 e PyQTGraph, que apresentam os dados que foram coletados e transmitidos via protocolo UDP pelo sensor em um gráfico, permitindo acompanhar em tempo real os impulsos musculares e com opção de armazenamento dos dados coletados. O sistema mostrou-se funcional, desde a aquisição até o armazenamento dos dados. Tal proposta se mostra promissora para estudos na área, principalmente como equipamento para fisioterapia, treinos esportivos e *biofeedbacks*.

Palavras-chave: Eletromiografia; Potencial de ação; Unidade motora; Eletrodo de Superfície; Sensor Eletromiográfico.

ABSTRACT

Electromyography (EMG) is an important area of studies in human physiology and health, encompassing the electrical current generated throughout the human body, from the cell membrane to muscle groups. A given electrical current is triggered in excitable membranes of muscle cells, which generates an action potential. A potential set, derived from cells that the same motor unit, results in a muscle contraction. The electromyograph is a device whose function is to capture the electrical potential of a muscle group, allowing monitoring, study and analysis of that group. Most current equipment has wires that are interconnected to the human body, hindering movement and still being a source of artifacts that generate noise that interferes with the EMG signal. Thus, the development of an electromyography system capable of performing the wireless acquisition, transmission and reception of electrical stimuli, transmitting them in a human-machine interaction interface, making it interesting and possibly viable for the current market. For the construction of the system, an electromyographic sensor was developed with an instrumentation amplifier, active filters, signal conditioning circuit and an ESP12F microcontroller, responsible for Wi-Fi acquisition and transmission through the signal's UDP protocol. A graphical interface was developed in Python, using the resources PyQT5 and PyQTGraph, which presents the data that were collected and transmitted via UDP protocol by the sensor in a graph, allowing to monitor in real time the muscular impulses and with the option of storing the collected data. The system proves to be functional, from acquisition to data storage. Such proposal is promising for studies in the area, mainly as equipment for physiotherapy, sports training and *biofeedbacks*.

Keywords: Electromyography; Surface Electrode; Action Potential; Electromyographic Sensor; Motor unit.

LISTA DE ILUSTRAÇÕES

Figura 1 – Camada bi-lipídica da membrana celular	15
Figura 2 – Bomba de sódio e potássio	16
Figura 3 – Potencial de Ação	17
Figura 4 – Somatório de potenciais de ação de uma unidade motora	18
Figura 5 – Exemplos de eletrodos	19
Figura 6 – Amplificador operacional ideal	21
Figura 7 – Arranjo interno de amplificadores de instrumentação (INA129)	21
Figura 8 – Comportamento ideal de um filtro passa-alta	23
Figura 9 – Comportamento ideal de um filtro passa-baixa	24
Figura 10 – Comportamento ideal de um filtro rejeita-faixa	24
Figura 11 – Camadas do modelo OSI	26
Figura 12 – Diagrama dos sistemas contidos no projeto	28
Figura 13 – Sinal eletromiográfico A original	29
Figura 14 – Sinal eletromiográfico B original	30
Figura 15 – Sinal eletromiográfico Agulha - Pré-tratamento para simulador	30
Figura 16 – Sinal eletromiográfico Superficial - Pré-tratamento para simulador	30
Figura 17 – Circuito do simulador	31
Figura 18 – Circuito de alimentação	32
Figura 19 – Circuito de transdução do sinal	33
Figura 20 – Circuito do filtro passa-alta	33
Figura 21 – Circuito do filtro passa-baixa	34
Figura 22 – Circuito do Amplificador Digital	34
Figura 23 – Circuito de Off-set	35
Figura 24 – Circuito do microcontrolador	36
Figura 25 – Fluxograma do Firmware	36
Figura 26 – Fluxograma de ações da interface	38
Figura 27 – Fluxo de Dados	38
Figura 28 – Imagem do simulador de sinal biológico montado em protoboard	39
Figura 29 – Sinal eletromiográfico A - Saída do conversor digital-analógico	39
Figura 30 – Sinal eletromiográfico do eletrodo B - Saída do conversor digital-analógico	40
Figura 31 – Sinal eletromiográfico A - Saída do simulador	40
Figura 32 – Sinal eletromiográfico B - Saída do simulador	41
Figura 33 – Sensor eletromiográfico desenvolvido	41
Figura 34 – Sinal eletromiográfico A - Saída do Amplificador Operacional Diferencial	42
Figura 35 – Sinal eletromiográfico B - Saída do Amplificador Operacional Diferencial	42
Figura 36 – Gráfico de bode do filtro passa-altas	43
Figura 37 – Gráfico de bode do filtro passa-baixas	43
Figura 38 – Espectro de sinal eletromiográfico	44
Figura 39 – Sinal A condicionado	45
Figura 40 – Sinal B condicionado	45
Figura 41 – Tela inicial da Interface	46
Figura 42 – Interface em funcionamento - plotagem do sinal A	47
Figura 43 – Interface em funcionamento - plotagem do sinal B	47
Figura 44 – Comparativo do Sinal eletromiográfico A	48

Figura 45 – Comparativo do sinal eletromiográfico B	49
Figura 46 – Sinal eletromiográfico B - Dados salvo em CSV	49
Figura 47 – Taxa de Pacotes recebidos por segundos	51

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AC	Corrente alternada
ADP	Adenosina difosfato
Amp op	Amplificador operacional
ATP	Adenosina trifosfato
DC	Corrente contínua
DAELN	Departamento acadêmico de eletrônica
EMG	Eletromiógrafo
I2C	Inter integrated circuit (Circuito Inter-Integrado)
IEC	Comissão eletrotécnica internacional
in amps	amplificador operacional de instrumentação
NBR	Norma brasileira
op amps	amplificador operacional
OSI	Open systems interconnect (Interconexão de sistemas abertos)
PTH	Pin Through Hole (Terminal inserido no furo)
SENIAM	Surface electromyograph for the non-invasive assessment of muscles
sEMG	Eletromiográfico de superfície
SMD	Surface Mount Device (Dispositivo montado em superfície)
TCP	Transmission control protocol (Protocolo de controle de transmissão)
UDP	User datagram protocol (Protocolo de datagramas de utilizador)
Wi-fi	Wireless Fidelity (Comunicação sem fio de alta fidelidade)

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVO	12
1.2	OBJETIVOS ESPECÍFICOS	12
1.3	JUSTIFICATIVA	12
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	POTENCIAL DE AÇÃO CELULAR	14
2.1.1	Membrana Celular	15
2.1.2	POTENCIAL DE AÇÃO	16
2.2	SINAL ELETROMIOGRÁFICO	18
2.2.1	AMOSTRAGEM DO SINAL	18
2.2.2	ELETRODOS	19
2.2.3	AMPLIFICADORES OPERACIONAIS	21
2.2.4	FILTROS	22
2.2.4.1	Filtro Passa-Alta	23
2.2.4.2	Filtro Passa-Baixa	23
2.2.4.3	Filtro Rejeita-faixa	24
2.2.5	RUÍDOS	24
2.3	COMUNICAÇÃO SEM FIO EM BIOSSINAIS	25
2.3.1	PROTOCOLO UDP	26
3	PROCEDIMENTOS METODOLÓGICOS	28
3.1	SIMULADOR DE SINAIS ELETROMIOGRÁFICOS	28
3.2	SENSOR ELETROMIOGRÁFICO	31
3.3	INTERFACE	36
4	RESULTADOS	39
4.1	SIMULADOR ELETROMIOGRÁFICO	39
4.2	SENSOR ELETROMIOGRÁFICO	41
4.3	INTERFACE	46
5	CONCLUSÃO	52
	REFERÊNCIAS	53
	APÊNDICE A – FIRMWARE DO ESP12F	57
	APÊNDICE B – CÓDIGO DA INTERFACE GRÁFICA	62

1 INTRODUÇÃO

A eletromiografia (EMG) é uma técnica que estuda a função muscular por meio da interpretação dos sinais bioelétricos emitidos pelo músculo (WEBSTER, 1998). Os eletromiógrafos podem ser utilizados na medicina e na fisioterapia, facilitando o acesso a processos fisiológicos e neuromusculares que acontecem no músculo durante a produção de força, sendo utilizados para diagnósticos de lesões, reeducação, treinamentos e *biofeedbacks* musculares (NODA *et al.*, 2014).

O sinal EMG captado é composto pelo somatório dos potenciais elétricos das células constituintes na unidade motora estudada. Os potenciais elétricos apresentam variação em sua amplitude, de acordo com o tamanho da unidade motora, assim, sendo possível capturar informações sobre sua atividade muscular (SOUZA, 2015). Existem duas principais formas de captação de sinais eletromiográficos: a forma invasiva, por eletrodos de profundidade (agulha); e a não invasiva, por eletrodos de superfície. A eletromiografia de superfície é amplamente utilizada na área da saúde, principalmente por ser um método não invasivo, o que facilita sua aplicação e manipulação em estudo de movimentos (BISCHOFF *et al.*, 1999).

Os aparelhos eletromiográficos convencionais possuem conectados a eles, por meios de fios e cabos, os eletrodos que são conectados aos pacientes e estes são ligados aos amplificadores e ao restante do sistema, o que pode resultar em erros de leitura devido a possíveis distorções de resultado, além de atrapalhar e gerar limitações na movimentação dos indivíduos estudados. Dessa forma, eletromiógrafos que possuem transmissão sem fio são uma alternativa para a medição de sinais musculares dinâmicos e que necessitam de um maior grau de liberdade em seus movimentos (OLIVEIRA, 2012).

Apesar do avanço tecnológico, a produção de módulos EMG nacionais ainda é pequena, sendo praticamente escassa do ponto de vista industrial, ocorrendo geralmente importações destes produtos. Ainda, são dispositivos limitados, capazes somente de amplificar e filtrar o sinal EMG (BECHELI, 2017). Deste modo, o desenvolvimento de um sistema capaz de realizar a transdução e condicionamento do sinal eletromiográfico para a transmissão sem fio, permitindo a visualização gráfica em tempo real dos sinais e armazenamento das informações obtidas por meio de uma interface amigável para o usuário pode ser de grande relevância para a área.

1.1 OBJETIVO

O objetivo do trabalho foi desenvolver um sistema de monitoramento eletromiográfico de superfície com comunicação sem fio, transmitindo os dados monitorados do grupo muscular para uma interface gráfica em tempo real com opção de armazenamento dos dados recebidos.

1.2 OBJETIVOS ESPECÍFICOS

- Desenvolver um protótipo funcional, integrando hardware, firmware e software de apresentação dos dados;
- Facilitar o processo de monitoramento de grupos musculares via EMG de superfície adicionando comunicação sem fio;
- Visualizar o sinal eletromiográfico em tempo real através de interface gráfica;
- Verificar funcionalidade do circuito condicionador de sinal;
- Utilizar bateria para eliminação de fios e ruídos provenientes deles e da rede elétrica.

1.3 JUSTIFICATIVA

O projeto surge como uma forma de estudo e desenvolvimento da área, visto que os aparelhos eletromiográficos mais acessíveis no mercado brasileiro ainda possuem um valor elevado, necessitando muitas vezes de importação. Além disso, estes equipamentos fazem a comunicação de seus componentes por fios o que impõem certas limitações no desenvolvimento de atividades dinâmicas durante o monitoramento de uma pessoa.

A presente pesquisa visa a contribuir para os estudos de sinais biológicos, visto que uma vez que captados com total liberdade de movimento pelo usuário, poderão ser salvos, revistos e comparados futuramente. Além disso, pode ser utilizada em aplicações nas áreas da saúde, como fisioterapia, permitindo seu uso em reabilitações, em acompanhando o desenvolvimento dos pacientes, em prevenções de lesões musculares, em biofeedbacks e auxílio em treinos esportivos, já que por ser portátil, possibilitará a reconstituição de movimentos naturais do atleta para estudo e avaliações de desenvolvimento.

Um atleta em sua rotina de treino poderá realizar movimentos idênticos ao que aplicaria em um jogo utilizando um eletromiógrafo sem fio, já que este não irá impor limitações em seus

movimentos, acompanhando seu quadro de força física, conseguindo aumentar os parâmetros de seu desenvolvimento. Um exemplo prático é a força exercida pelo ombro de um jogador de vôlei para a realização de saque com salto ou a força de impulsão nas panturrilhas e coxas de um jogador de handebol na execução de um arremesso em suspensão, que poderão ser medidas durante um treino com bola, podendo identificar os limites físicos de cada jogador, a compensação de força muscular realizada pelo corpo e o acionamento correto da musculatura, arquivando tais dados para análise futura e verificação de sua evolução.

Becheli (2017) e Alasker (2014) ainda demonstram em seus trabalhos que a eletromiografia é uma área com estudos em ascensão, tanto para integração computacional dos sinais captados, quanto para utilização de biosinais para próteses robóticas para reabilitação.

2 FUNDAMENTAÇÃO TEÓRICA

O estudo das funções musculares através do potencial elétrico manifestante associado a um grupo muscular é realizado através da eletromiografia (WEBSTER, 1998). Esse estudo monitora os sinais bioelétricos, derivados de membranas celulares excitáveis, por uma corrente gerada através do fluxo iônico entre as membranas das fibras musculares que se propagam até o ponto de detecção por um eletrodo. O sinal do EMG é o somatório de todos os potenciais musculares circundantes a área estudada (NODA *et al.*, 2014; LUCA, 2006).

Existem várias aplicações para o sinal de EMG, desde diagnósticos até acompanhamento de diversas patologias. Segundo Siqueira Junior (2013), Luca (2002) e Velloso (2004) pode-se afirmar que a partir do sinal captado é possível (I) aplicar análises para determinar o tempo de ativação muscular (quando a excitação muscular inicia e termina), (II) avaliar a velocidade de condução de fibras musculares, (III) estimar a força produzida pelo grupo muscular, (IV) identificar a taxa de fadiga do músculo através de análise em frequência, (V) avaliar o trabalho de um grupo muscular, (VI) controlar próteses de membros artificiais, (VII) diagnosticar problemas neuromotores, (VIII) auxiliar na medicina comportamental e ergonômica, e (IX) realizar a aplicação em técnicas de *biofeedback*, que consiste no monitoramento de diversas funções involuntárias do corpo, de modo a permitir o indivíduo a desenvolver a capacidade de autorregulação.

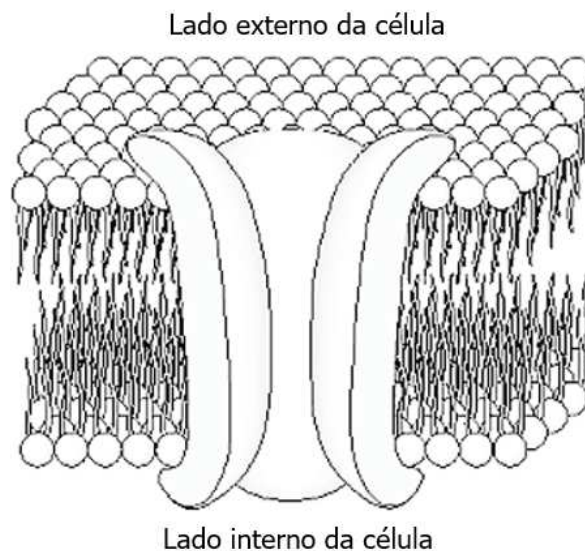
2.1 POTENCIAL DE AÇÃO CELULAR

Todas as células que compõem os seres vivos possuem uma membrana que as delimitam, separado o meio extra do intracelular. Todas as células possuem um potencial elétrico sobre sua membrana celular, mas apenas algumas permitem a alteração deste potencial elétrico sobre a membrana para a transmissão de informação. Portanto, a variação da tensão elétrica na célula desempenha uma ação fundamental no controle das funções celulares, e conseqüentemente, no controle vital dos organismos.

2.1.1 Membrana Celular

A membrana celular é constituída por duas camadas de ácidos graxos fosfoglicerídeos, que possuem em uma extremidade da molécula, ácido fosfórico, dotados de propriedades hidrofílicas, e em sua outra extremidade, uma espécie de calda molecular composta por uma cadeia de glicerídeos com propriedades hidrofóbicas. Estas moléculas formam um aglomerado laminar em dupla camada com espessura de 7,5-10 nm, posicionando a parte hidrofóbica em sua parte externa e a hidrofílica interna. Possuem também poros proteicos que formam canais iônicos, tornando-a seletivamente permeável para o fluxo de sódio, potássio e cloro através da membrana (MALMIVUO; PLONSEY, 1995). A Figura 1 ilustra o posicionamento dos ácidos graxos em dupla camada.

Figura 1 – Camada bi-lipídica da membrana celular

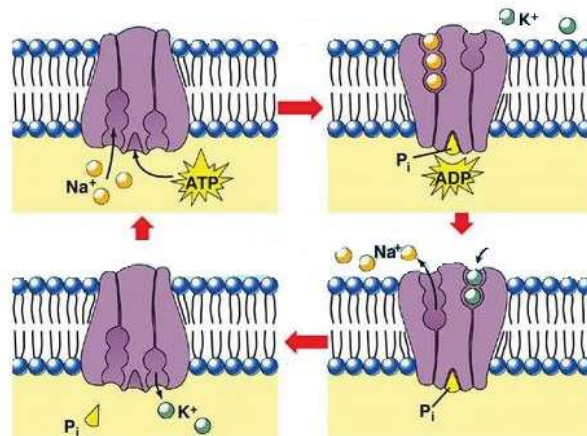


Fonte: Malmivuo e Plonsey (1995).

A permeabilidade seletiva da membrana celular é possível devido a existência de proteínas que atuam como canais iônicos, que permitem a passagem dos íons de sódio (Na^+) e potássio (K^+). Porém estes canais são 100 vezes mais permeáveis para o K^+ do que para Na^+ . A bomba de sódio e potássio é o mecanismo responsável pela manutenção do estado de desbalanceamento iônico entre o meio extracelular e o intracelular, realizando transporte ativo de íons contra seu respectivo gradiente de concentração, carregando continuamente três íons de sódio para o exterior celular e simultaneamente carregando dois íons de potássio para o interior celular, utilizando a molécula adenosina trifosfato (ATP) como fonte de energia para esse

processo, transformando ela em adenosina difosfato (ADP) (OLIVEIRA, 2012). Este fenômeno encontra-se ilustrado na Figura 2.

Figura 2 – Bomba de sódio e potássio



Fonte: Magalhães (2017).

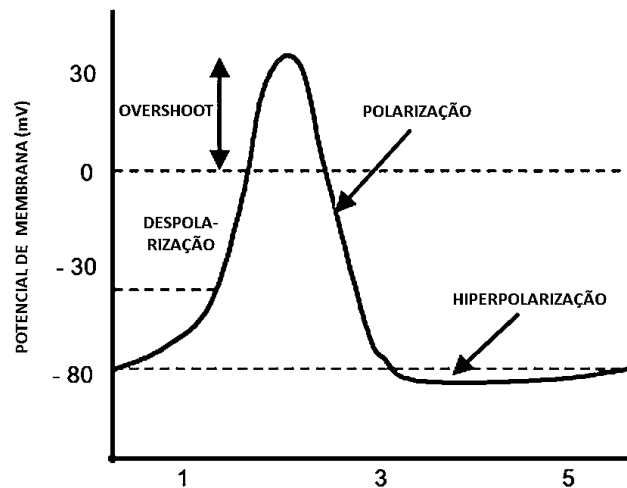
2.1.2 POTENCIAL DE AÇÃO

A diferença de íons entre o meio extracelular e intracelular resulta em uma diferença de potencial elétrico, conhecido como potencial de membrana. O potencial de membrana para a maioria das células fica entre -60 e -90 mV, sendo conhecido como potencial de *Nerst*. O interior da célula possui potencial negativo em relação ao meio externo, ocorrendo variações como resultado das trocas iônicas dos fluídos celulares. Essas variações também podem ocorrer devido a excitabilidade que a membrana celular possui, principalmente células nervosas e musculares, podendo chegar a aproximadamente +20 ou +50 mV em poucos milissegundos, sendo denominado potencial de ação (RODRIGUEZ-AÑEZ, 2000).

O potencial de ação é caracterizado como uma rápida variação do potencial de membrana, ou rápida inversão de polaridade da membrana, passando de um valor negativo em seu estado de repouso, dito polarizado, para um número que varia de seu valor inicial em estado estacionário para até +30 mV em seu estado excitatório, gerando um estado de despolarização na membrana celular. Quando a magnitude da tensão elétrica sobre a membrana é mais negativa que seu potencial de repouso, é dito que esta está hiperpolarizada. Devido a essa reversão de polaridade, cada ciclo de potencial de ação possui uma fase ascendente, outra descendente, seguida de uma hiperpolarização e volta ao potencial elétrico de repouso de membrana (IDE *et al.*, 2012), como mostrado na Figura 3 . Estas variações de tensão sobre a membrana permitem a

existência do potencial de ação, fenômeno fundamental para a transmissão de informação na maioria dos organismos vivos (OLIVEIRA, 2012).

Figura 3 – Potencial de Ação



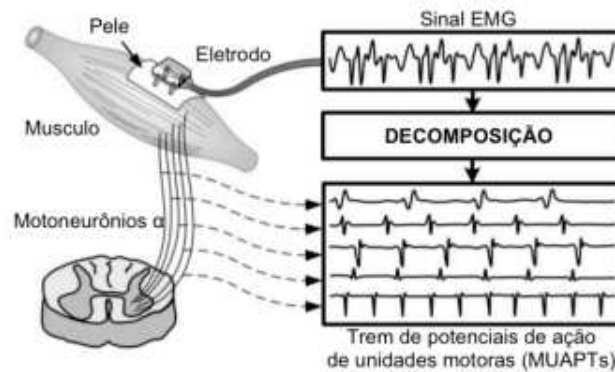
Fonte: Adaptado de Konrad (2005).

Neste processo de geração do potencial de ação, a quantidade de íons transportados durante a difusão não é relevante, se comparado às concentrações nos meios intra e extracelulares durante a excitabilidade das células, dessa forma, o fluxo de íons não exerce efeito sobre as concentrações totais, e após o início do potencial, outro só terá início após o término do primeiro, não ocorrendo a sobreposição de potenciais. (IDE *et al.*, 2012).

As ramificações dos terminais neuromotores são excitadas por um potencial de ação proveniente do sistema nervoso, este sinal por sua vez é transferido para as fibras musculares por meio das placas neurais localizadas, em grande parte, aproximadamente na porção central da fibra. O potencial de ação percorre os dois sentidos da fibra e termina em ambas as extremidades, percorrendo a fibra muscular de forma a desencadear a contração muscular (OLIVEIRA, 2012).

Durante as contrações, a força pode ser modulada por uma combinação de recrutamento e alternância de unidades motoras ativas, sendo que quanto maior a frequência dos disparos e o número de unidades motoras recrutadas, maior será a força gerada (IDE *et al.*, 2012). Dessa forma, entende-se que a unidade motora é formada por várias fibras musculares, e que a magnitude do potencial de ação celular incorporados nas fibras inervadas dentro desta unidade motora, formam um somatório de potenciais de ação da unidade, resultando em um potencial de maior magnitude, conforme o tamanho da unidade motora (KONRAD, 2005), como mostrado na Figura 4.

Figura 4 – Somatório de potenciais de ação de uma unidade motora



Fonte: SIQUEIRA JUNIOR (2013).

2.2 SINAL ELETROMIOGRÁFICO

Com o uso da eletromiografia é possível analisar um grupo muscular ou determinado feixe muscular, através da aquisição dos impulsos musculares que provêm os sinais, possibilitando análise de duração e a intensidade (NODA *et al.*, 2014).

Cada potencial de ação contribui para a formação de um sinal que é captado pelos eletrodos posicionados em uma determinada unidade motora. Este sinal depende da distância entre a fibra muscular e a área onde estão posicionados os eletrodos, das características elétricas dos tecidos em volta de cada fibra muscular, do diâmetro das fibras, da distância entre os eletrodos e suas propriedades (OLIVEIRA, 2012). As fibras musculares de uma unidade motora estão espalhadas aleatoriamente por uma seção muscular e entrelaçadas com fibras de outras unidade motoras. Assim, o sinal resultante captado pelo eletrodo de superfície é a soma algébrica de diversos potenciais de ação das unidades motoras (OLIVEIRA, 2012). É estabelecido que a amplitude do sinal EMG é de natureza estocástica (aleatória) e pode ser razoavelmente representada por uma função de distribuição gaussiana (LUCA, 2002).

2.2.1 AMOSTRAGEM DO SINAL

O sinal eletromiográfico possui baixa amplitude e uma vasta banda de passagem. Segundo Bischoff *et al.* (1999), Siqueira Junior (2013) e Luca (2002), a amplitude do sinal pode variar de 1 mV até 10 mV (pico a pico) ou 0 a 1,5 mV (rms), contendo componentes de até 10 kHz. Becheli (2017), Merletti *et al.* (2008) e Oliveira (2012) ainda ressaltam que a maior energia do sinal do EMG está contido entre 5Hz e 500 Hz, e segundo Hermens *et al.* (2012) do projeto

européu SENIAM (*Surface Electromyography for the Non-Invasive Assessment of Muscles*), as harmônicas de até 400 Hz significam 95% da energia do sinal, sendo a harmônica entre 400 Hz e 450 Hz, a última harmônica de interesse.

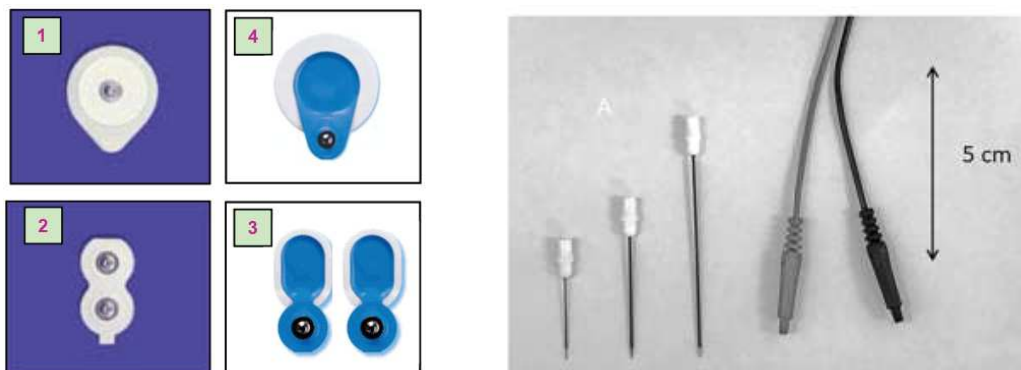
Para uma boa amostragem do sinal deve-se seguir o teorema de *Nyquist*, que foi desenvolvido por Harry Nyquist na década de 1920, e que determina que para um sinal ser reconstituído a partir de suas amostras, sua frequência de amostragem deve ser no mínimo duas vezes maior do que a frequência mais alta encontrada no sinal. Dessa forma, quanto mais alto a taxa de amostragem, maior será a fidelidade do sinal reconstruído (VELLOSO, 2004).

Segundo Konrad (2005), utilizando uma frequência até 500 Hz é suficiente para a captação do sinal de EMG superficial. Dessa forma, será necessário realizar uma amostragem de pelo menos 1 kHz, mas para obter um sinal fidedigno, Chauvet *et al.* (2003) recomenda a realização de amostragens de aproximadamente 10 kHz.

2.2.2 ELETRODOS

Existem dois principais tipos de eletrodos para a eletromiografia, os de superfície e os de profundidade, ambos possuindo grande variedade de formas e materiais. Os eletrodos de superfície são considerados não invasivos, visto que estes são simplesmente colocados sobre a pele, enquanto eletrodos de profundidade são considerados invasivos, sendo geralmente eletrodos do tipo agulha em contato direto com o tecido muscular do indivíduo (RODRIGUEZ-AÑEZ, 2000). Na Figura 5 pode-se verificar alguns modelos de eletrodos de superfície em (a) e de eletrodos de profundidade em (b).

Figura 5 – Exemplos de eletrodos



(a) Tipos de eletrodos de Superfície.

(b) Tipos de eletrodos de profundidade.

Fonte: a) Adaptado de Konrad (2005). b) Adaptado de Stalber *et al.* (2019).

Os métodos invasivos trazem maior qualidade no procedimento, maior segurança e precisão nas informações colhidas, pois captam a atividade de um menor número de unidades motoras (NODA *et al.*, 2014). Contudo, os eletrodos de profundidade trazem maior desconforto ao paciente na etapa de colocação, além de possuírem maior custo e de uma maior complexidade do exame. Dessa forma os não-invasivos são mais utilizados (GARCIA *et al.*, 2004).

Os eletrodos de superfície podem ser definidos como um sensor de atividade mecânica de uma unidade motora ou como um transdutor da corrente iônica para corrente elétrica. Um eletrodo de superfície é caracterizado por sua dimensão física, forma, tecnologia e materiais constituintes, todos esses fatores podem influenciar a aquisição do sinal EMG (FORTI, 2005).

Considerando os materiais e tecnologias utilizadas para a fabricação de eletrodos de superfície, pode-se distingui-los em secos e não secos. Os eletrodos são constituídos de metais nobres (ouro, prata ou platina), de carbono, de prata sinterizada ou de prata-cloreto de prata (Ag/AgCl). Contudo, o que diferencia ambos é que os eletrodos úmidos incluem uma camada de gel condutor, hidrogel ou esponja saturada como eletrólito, que funcionam como interface entre os eletrodos e a pele do indivíduo. Estes geralmente são autoadesivos (MERLETTI *et al.*, 2008).

Os eletrodos de Ag/Agcl de superfície se sobrepõem aos demais materiais em seu uso devido ao seu baixo potencial de meia-célula, que envolve as reações eletroquímicas no eletrodo, baixo desvio do potencial de difusão, insensibilidade à temperatura (NARAYAN *et al.*, 2015), alta estabilidade, apresenta menor nível de ruído na junção com o gel em relação aos outros eletrodos metálicos, possui fácil manuseio e baixo custo (MERLETTI *et al.*, 2008).

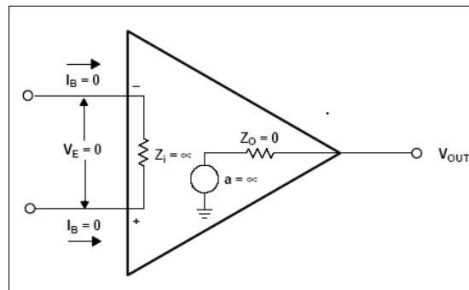
Pode-se ainda distinguir os eletrodos de superfície em monopolares ou bipolares. Os monopolares possuem apenas um eletrodo sobre o grupo muscular de interesse e outro sobre um ponto que não é afetado pela atividade daquele grupo muscular, chamado de referência, captando assim a diferença de potencial entre os dois pontos. Já os bipolares consistem de dois eletrodos sobre o grupo muscular estudado e um de referência, medindo a diferença de potencial entre os dois (RODRIGUEZ-AÑEZ, 2000).

O tamanho dos eletrodos possui relevância no estudo, pois eles devem ser grandes o suficiente para cobrir parte do conjunto da unidade motora a ser monitorada, mas também pequeno o suficiente para evitar o *cross-talk*, fenômeno conhecido por captar interferências dos músculos ao redor. Para monopolares são recomendados eletrodos até 10 mm e para bipolares até 20 mm de diâmetro (SIQUEIRA JUNIOR, 2013).

2.2.3 AMPLIFICADORES OPERACIONAIS

Os amplificadores operacionais, também conhecidos como *op amp*, são amplificadores diferenciais que possuem um ganho muito alto, impedância de entrada muito alta e baixíssima impedância de saída (BOYLESTAD, 2012). Os *op amps* são utilizados na construção de buffers, filtros ativos, amplificadores inversores, amplificadores não inversores, comparadores, somadores, integradores e amplificadores de instrumentação, sendo a base dos circuitos de condicionamento de sinais analógicos (VELLOSO, 2004). Na Figura 6 é mostrado um amplificador operacional ideal.

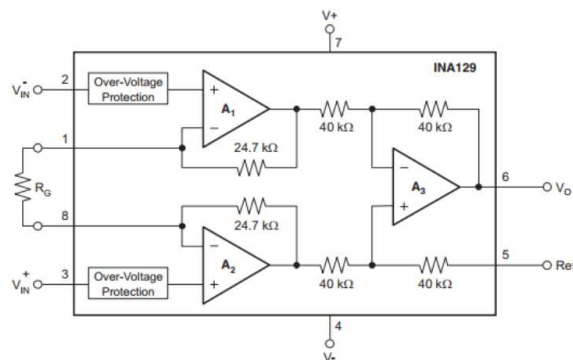
Figura 6 – Amplificador operacional ideal



Fonte: Malvino e Bates (2016).

A partir dos amplificadores operacionais é construído o amplificador de instrumentação diferencial, também conhecido com *in amps*. Para a construção destes dispositivos são utilizados três *op amps* no circuito, resultando em uma alta taxa de rejeição de modo comum (CMRR) e impedâncias de entrada próximas uma da outra, obtendo maior precisão na aquisição do sinal (VELLOSO, 2004). Na Figura 7 é demonstrado um circuito integrado INA129 da fabricante TEXAS INSTRUMENTS como exemplo de estrutura interna de *in amps*.

Figura 7 – Arranjo interno de amplificadores de instrumentação (INA129)



Fonte: TEXAS TEXAS INSTRUMENTS (2015).

Algumas características adicionais compõem o amplificador de instrumentação que o diferenciam do amplificador operacional convencional, que incluem um off-set DC muito baixo, alta imunidade a ruído, um ganho alto de malha aberta, altíssima impedância de entrada, elevada taxa de rejeição em modo comum e recuperar-se rapidamente após a saturação (BECHELI, 2017; BISCHOFF *et al.*, 1999).

O amplificador de instrumentação diferencial tem por função amplificar a diferença de potencial entre suas entradas, porém estas não são ideais, resultando que sinais de modo comum acabem por aparecer na saída do amplificador, mas atenuadas pela taxa de relação de rejeição em modo comum (CMRR), por isso *in amps* devem possuir CMRR de no mínimo 90 dB (OLIVEIRA, 2012). Para Siqueira Junior (2013) e Merletti *et al.* (2008), o ruído em um sinal de saída de eletromiografia deve ser menor que 1 μV_{rms} , ou seja, três ordens menor que a magnitude do sinal de sEMG, tornando sua energia desprezível.

Este tipo de amplificador é especialmente útil, devido as suas características, na aquisição de sinais reduzidos e quando a interferência se sobrepõe de forma significativa ao sinal, fato que ocorre frequentemente na aquisição de biossinais.(VELLOSO, 2004)

2.2.4 FILTROS

Os filtros aplicados em sinais biológicos possuem três funções principais, que são: (I) a redução de ruídos, (II) a supressão de artefatos e potenciais interferentes na parte desejada do sinal e (III) prover informações de um sinal que podem não ser intuitivas sob condições comuns (BISCHOFF *et al.*, 1999).

Para a captação de sinais de EMG, normalmente é utilizado primeiramente um filtro passa-alta, para eliminação de ruídos providos de artefatos de movimentação, seguido de um filtro passa-baixa para limitação da banda de passagem do sinal, e finalmente um filtro *notch* para redução das interferências da rede elétrica (OLIVEIRA, 2012).

Na implementação dos filtros, a ordem deles é de grande relevância, pois quanto maior a ordem, mais próximo estarão do ideal, melhorando sua resposta e atenuação das frequências indesejadas do sinal (VELLOSO, 2004).

Os filtros digitais também são uma poderosa ferramenta para a realização de filtragens de sinais, pois a realizam de forma semelhante aos filtros analógicos implementados em *hardware*, mas sem apresentar problemas relacionados a imperfeições dos componentes, variações de temperatura, etc., porém ainda devem existir filtros analógicos limitando a banda de passagem

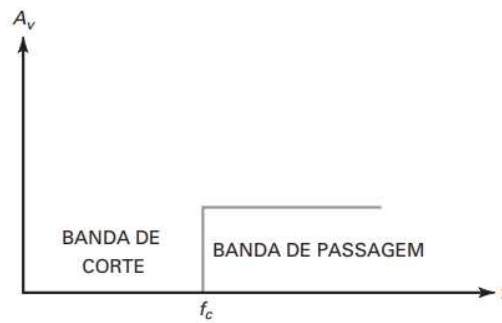
para digitalização do sinal (VELLOSO, 2004).

2.2.4.1 Filtro Passa-Alta

Os filtros passa-alta atenuam as baixas frequências, agindo desde um pulso contínuo até a frequência de corte, permitindo a passagem de todas as frequências acima da frequência de corte (MALVINO; BATES, 2016), como demonstrado em seu comportamento ideal pela Figura 8.

Na eletromiografia, o filtro passa-alta é comumente utilizado para a eliminação de ruídos gerados pela movimentação mecânica do sensor e componentes de tensão contínua que podem se originar na interface entre pele e eletrodo (OLIVEIRA, 2012).

Figura 8 – Comportamento ideal de um filtro passa-alta



Fonte: Malvino e Bates (2016).

2.2.4.2 Filtro Passa-Baixa

Os filtros passa-baixa atenuam as altas frequências, permitindo a passagem de todas as frequências desde zero até a frequência de corte e bloqueando todas as frequências acima da frequência de corte (MALVINO; BATES, 2016), tendo um comportamento ideal conforme Figura 9.

Um filtro passa-baixa na eletromiografia é utilizado como limitador de banda de frequência e para remoção de ruídos de alta frequência (OLIVEIRA, 2012).

Figura 9 – Comportamento ideal de um filtro passa-baixa



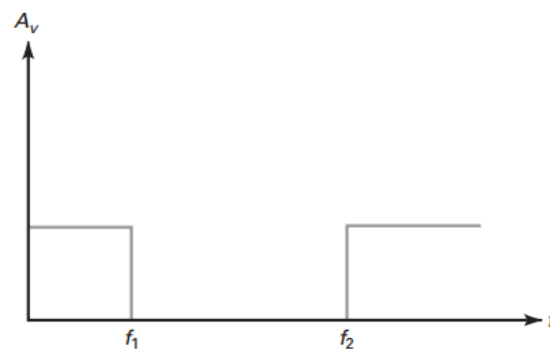
Fonte: Malvino e Bates (2016).

2.2.4.3 Filtro Rejeita-faixa

Os filtros rejeita-faixa, também conhecidos como *notch*, permitem a passagem de todas as frequências desde zero até a frequência de corte inferior (f_1), bloqueia todas as frequências entre as frequências de corte inferior (f_1) e superior (f_2), e permite a passagem de todas as frequências acima da frequência de corte superior (f_2) (MALVINO; BATES, 2016), tendo um comportamento ideal conforme Figura 10.

Um filtro rejeita faixa é utilizado na eletromiografia para eliminar frequências de interferência da rede elétrica presentes no sinal, principalmente a frequência fundamental de 60 Hz (OLIVEIRA, 2012).

Figura 10 – Comportamento ideal de um filtro rejeita-faixa



Fonte: Malvino e Bates (2016).

2.2.5 RUÍDOS

A faixa de amplitude de sinal que um eletromiógrafo é capaz de detectar variações de poucos μV até mV , tornando o equipamento propenso à captação de ruídos externos (OLIVEIRA,

2012). Assim, um sistema de eletromiografia pode sofrer diferentes maneiras de interferência, podendo ser (I) por indução elétrica e magnética, (II) correntes de deslocamento nas derivações do eletrodo, (III) correntes de deslocamento no corpo e (IV) interconexão e imperfeições de equipamentos (HUHTA; WEBSTER, 1973).

As interferências podem ser providas das mais diversas fontes, ocasionando um vasto espectro de frequência de ruído no sinal. Por ondas eletromagnéticas, tem-se ondas de rádio e televisão, que ocorrem em alta frequência (kHz e MHz), interferências da rede elétrica, e, como essa está presente em nosso cotidiano, é inevitável a aparição de ruídos de 50/60 Hz em aparelho conectados na rede (BECHELI, 2017).

Já o problema na aquisição de sinais com eletrodos, ocorre com os cabos que interligam estes ao circuito, devendo os cabos estarem o mais próximos possível uns dos outros para que a interferência captada em cada um deles seja a mesma e possa ser propriamente eliminada pela amplificação diferencial, pois caso a interferência em cada cabo seja distinta, acabará por ser amplificada esta diferença entre a interferência e o cabo, ocasionando perda de qualidade no sinal desejado (VELLOSO, 2004).

Pode-se ainda ter interferências provindas de músculos vizinhos ao estudado, que podem produzir uma quantidade significativa de EMG que é detectada pelo eletrodo. Normalmente, o *cross talk* não excede 10% a 15% do sinal do músculo estudado. Os picos de eletrocardiograma também podem interferir no EMG, principalmente quando realizadas captações nos músculos superiores, como zona peitoral, tronco e ombro (KONRAD, 2005).

A interferência também pode ocorrer por imperfeições entre interconexões eletromecânicas, como a interface entre a região de detecção do eletrodo e a pele, que em conjunto com a metodologia de posicionamento e fixação (limpeza, tipo de gel, sentido da fibra muscular, etc.) podem originar instabilidade no sinal que ocupa uma faixa de passagem de 0 a 20Hz (SIQUEIRA JUNIOR, 2013).

2.3 COMUNICAÇÃO SEM FIO EM BÍOSSINAIS

O desenvolvimento da tecnologia a partir da década de 1990 foi um marco no desenvolvimento de hardware e software para uso pessoal. Os computadores pessoais passaram a se apresentar capazes de lidar com processos relacionados a gravação, capacidade de transferir dados, processar sinais rapidamente, apresentação gráfica versátil, plataforma de armazenamento de dados compreensível e capaz de gerar relatórios, fazendo mudar o conceito de integração

entre aparelhos eletromiográficos e computadores pessoais (LADEGAARD, 2010). Portanto, os eletromiógrafos passaram a se integrar com computadores pessoais.

A construção de eletromiográficos com sistemas de transmissão de dados sem fio está permitindo exames com o paciente em movimento, totalmente livre de fios, abrindo novas possibilidades para esta tecnologia e estudos.

2.3.1 PROTOCOLO UDP

O modelo OSI (*Open Systems Interconnect*) foi criado como padrão para protocolos de comunicação entre diversos tipos de sistemas, entre máquinas distintas e dispositivos genéricos independentemente da tecnologia utilizada, permitindo assim compatibilidade, portabilidade, interoperabilidade e escalabilidade do uso de redes. Este modelo tem suas funções divididas em sete camadas de abstração com cada camada possuindo uma função específica, sendo elas as camadas de física, enlace, rede, transporte, sessão, apresentação e aplicação (SILVEIRA, 2016), estando dispostas conforme ilustra a Figura 11. Nas camadas 1, 2 e 3 ocorrem a interface de rede enquanto nas camadas 5, 6 e 7 operam próximo a interface de sistema, e a camada 4 tem a função de fazer o intermédio entre as informações dessas interfaces, realizando o transporte de informações entre máquinas.

Figura 11 – Camadas do modelo OSI

7	CAMADA DE APLICAÇÃO	}	APLICAÇÃO
6	CAMADA DE APRESENTAÇÃO		
5	CAMADA DE SESSÃO		
4	CAMADA DE TRANSPORTE	}	REDE
3	CAMADA DE REDE		
2	CAMADA DE ENLACE		
1	CAMADA FÍSICA		

Fonte: Adaptado de Silveira (2016).

A camada de transporte visa garantir que as mensagens sejam entregues sem erros, em sequência, sem perdas e sem duplicações, eliminando para os protocolos de camadas superiores qualquer preocupação a respeito de transferência de dados entre eles. Existem vários protocolos

que podem ser utilizados para esta camada, mas o tamanho e a complexidade do protocolo dependerá do tipo de serviço que ele pode obter da interface de rede (SILVEIRA, 2016). Os dois principais protocolos utilizados nesta camada é o TCP (*Transmission Control Protocol*) e o UDP (*User Datagram Protocol*) (ORESTES, 2019).

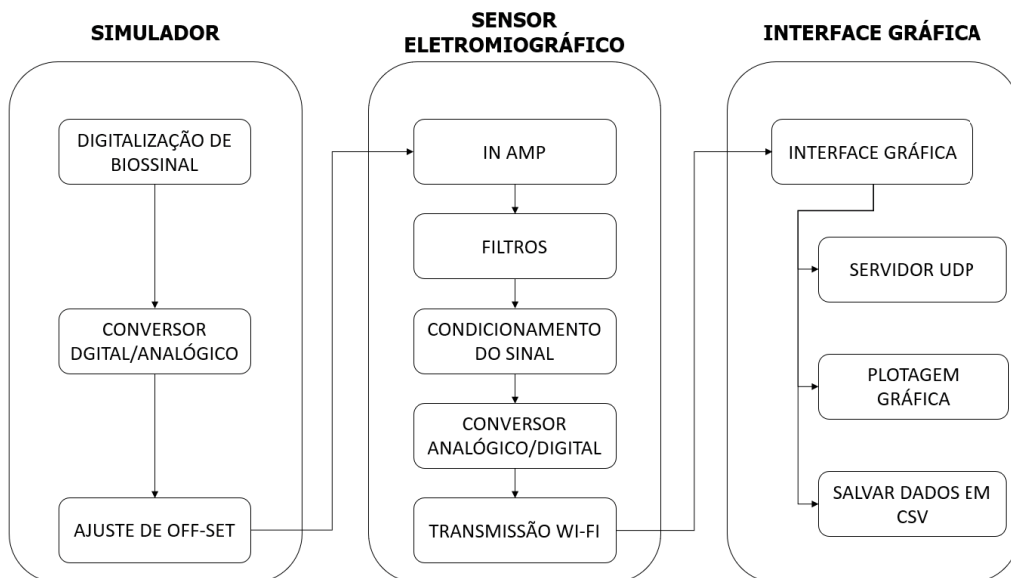
Os protocolos TCP e UDP se baseiam no envio de pacotes de informações, sendo que a principal diferença entre eles é que o UDP remove toda a parte de verificação de erros enquanto o TCP o mantém. O objetivo dessa opção é acelerar o processo de envio de dados, visto que todas as etapas de comunicação necessárias para verificar a integridade de um pacote (e para reenviá-lo, se necessário) contribuem para deixar mais lento (GUGELMIN, 2014).

Ao utilizar o protocolo UDP, acaba-se por simplesmente mandar informações a um destinatário, sem verificar se elas foram recebidas devidamente, e caso ocorram erros, ocorrerá o envio do próximo pacote programado pelo sistema, e os anteriores não serão recuperados. Esse método de funcionamento aumenta a ocorrência de erros, mas melhora o desempenho de aplicações que não precisam ou podem esperar pelo recebimento de um pacote que se perdeu, sendo fator importante para sistemas de tempo real (GUGELMIN, 2014).

3 PROCEDIMENTOS METODOLÓGICOS

Para desenvolver o sistema eletromiográfico sem fio necessitou-se de dois sistemas que interagem entre si, sendo eles o sensor eletromiográfico com transmissor sem fio, que realizará a aquisição do sinal biológico, seu condicionamento e o transmitirá via Wi-Fi e a interface cujas funções são interação homem-máquina e salvar os dados obtidos. Para a realização dos teste do sistemas eletromiógrafo, desenvolveu-se um sistema simulador de sinal biológico para reprodução de um sinal eletromiográfico real, fornecendo assim o sinal que o sensor eletromiográfico realizará leitura e o enviará para a interface. Na figura abaixo é possível visualizar os sistemas e as funções que possuem.

Figura 12 – Diagrama dos sistemas contidos no projeto



Fonte: Autoria própria (2021).

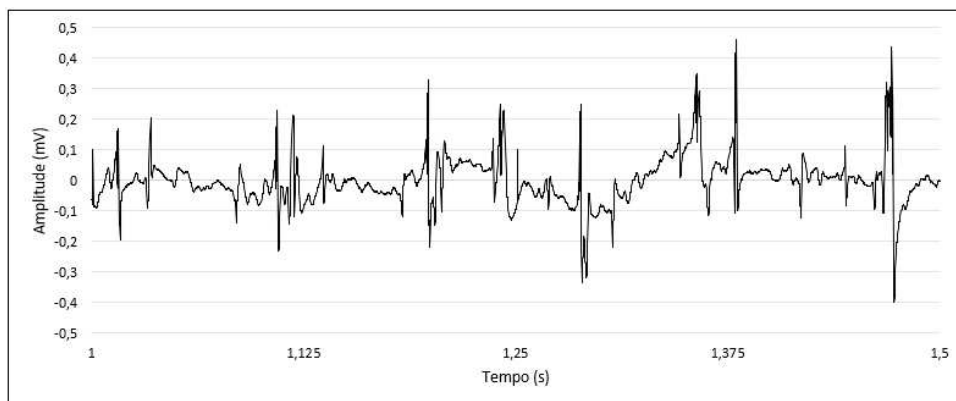
3.1 SIMULADOR DE SINAIS ELETROMIOGRÁFICOS

Para desenvolvimento do simulador utilizou-se como base dois sinais eletromiográficos obtidos dos bancos de dados Physiobank e NinaPro, que serão denominados sinal A e sinal B. O sinal A foi escolhido por ter sido coletado a partir de eletrodo agulha (invasivo), possuindo seu sinal bem definido, já que foram retirados de um único músculo; já o sinal B, foi escolhido por utilizar eletrodos de superfície (não-invasivo), sendo seu sinal composto pelo somatório dos impulsos elétricos de todos os músculos que compõem o grupo muscular, sendo a situação real

de um eletromiógrafo de superfície.

Segundo as informações apresentadas em PHYSIONET (2016), para obtenção do sinal A, o eletrodo invasivo utilizado foi uma agulha concêntrica de 25 mm colocada no músculo tibial anterior de um homem de 44 anos sem histórico de doença neuromuscular utilizando um sistema de monitoramento Medelec Synergy N2 EMG (Oxford Instruments Medical, Old Woking, Reino Unido), os dados foram gravados em 50 kHz e então diminuídos para 4 kHz. Durante o processo de gravação foram usados dois filtros analógicos, um filtro passa-alta de 20 Hz e um filtro passa-baixa de 5KHz. Foram coletados 12 segundos de dados, sendo a faixa selecionada para utilização apenas a entre 1 a 1,5 segundos, conforme gráfico apresentado na Figura 13.

Figura 13 – Sinal eletromiográfico A original

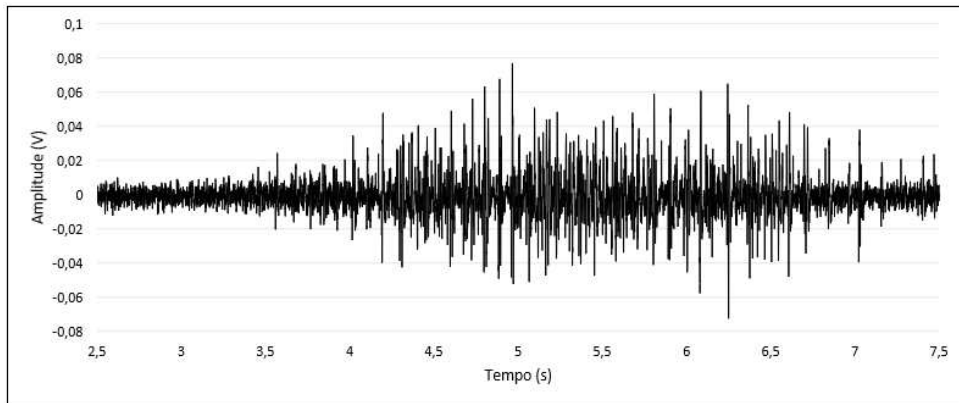


Fonte: Autoria própria (2021).

Já o sinal B, seguindo as informações em IEE-LAB (2017), foi adquirido através de um eletromiógrafo comercial, EMG 830 C, da EMG System do Brasil, com sinal amostrado a 2 kHz e digitalizado em 18 bits por uma plataforma NI USB-6289 da National Instruments, sendo adquirido por um notebook executando o *software* LabVIEW. As amostras foram coletadas com os eletrodos posicionados em 12 pontos do braço, reproduzindo os 17 movimentos de mão e punho sugeridos pelo banco de dados NINAPRO (IEE-LAB, 2017). Utilizou-se o sinal obtido pelo eletrodo posicionado sobre o bíceps do voluntário. Este sinal foi coletado enquanto o indivíduo, sentado conformavelmente, realizava vários movimentos, optando assim por utiliza o sinal entre a faixa de 2,5 a 7,5 segundos para os testes neste trabalho, conforme apresentado na Figura 14.

Ambos os sinais receberam pré-tratamento para poderem ser utilizados no circuito simulador. O sinal A devido sua baixíssima amplitude foi amplificado em 10 vezes, obtendo amplitude correspondente a um sinal de eletromiografia de superfície. Então para ambos sinais,

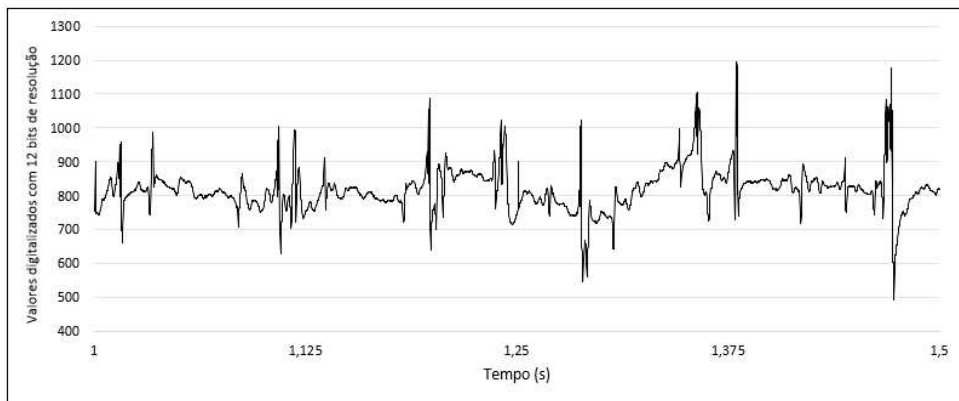
Figura 14 – Sinal eletromiográfico B original



Fonte: Autoria própria (2021).

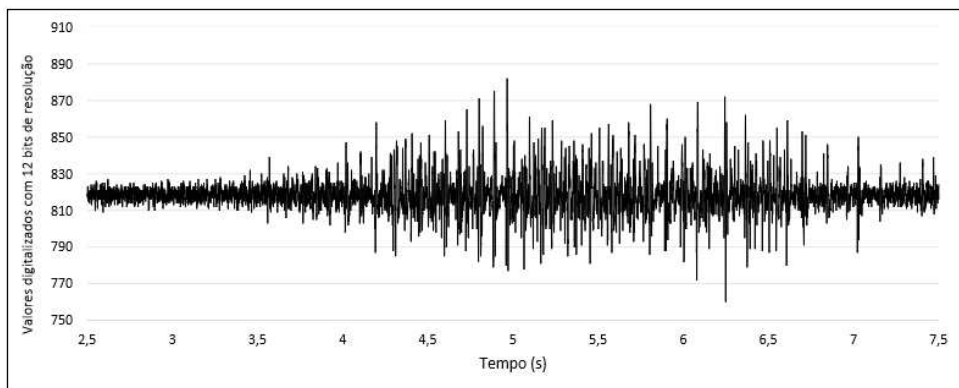
onde possuíam componentes positivas e negativas de tensão, somou-se 1 V em suas amplitudes, que passaram a possuir apenas componentes positivas em seus sinais, sendo posteriormente digitalizados para 12 bits, obtendo assim os sinais conforme Figura 15 e Figura 16.

Figura 15 – Sinal eletromiográfico Agulha - Pré-tratamento para simulador



Fonte: Autoria própria (2021).

Figura 16 – Sinal eletromiográfico Superficial - Pré-tratamento para simulador

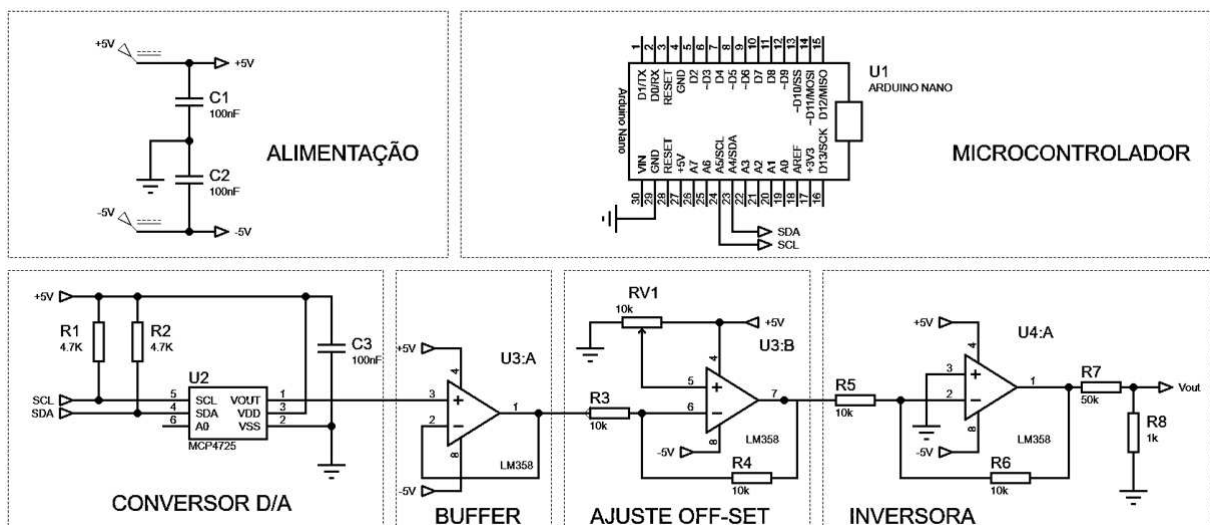


Fonte: Autoria própria (2021).

Com o sinal digitalizado, o mesmo foi gravado através da IDE Arduino em um microcontrolador ATmega328-PU contido em uma placa de desenvolvimento Arduino Nano. Através da comunicação I2C (*Inter-Integrated Circuit* - barramento serial desenvolvido pela *Philips*) o sinal eletromiográfico foi transmitido do ATmega328 para um circuito integrado MCP4725, que tem por função converter um sinal digital para analógico com uma resolução de 12 bits (MICROCHIP, 2007), respeitando a taxa de amostragem original de cada sinal utilizado (4 kHz para o sinal A e 2 kHz para o sinal B). Este conversor é muito indicado para projetos de instrumentos musicais, gerador de sinais e outras aplicações (BORGES, 2018).

Após a saída do MCP4725, tem-se um buffer e então um circuito de ajuste de off-set, com a função de remover a tensão que foi adicionada anteriormente no vetor dos sinais eletromiográficos, fazendo com que estes sinais voltem a possuir componentes positivas e negativas, por fim um circuito inversor seguido de um divisor de tensão fazem que o sinal seja ajustado de forma adequada a se aproximar do original. A Figura 17 mostra o circuito completo do simulador de sinais de EMG.

Figura 17 – Circuito do simulador



Fonte: Autoria própria (2021).

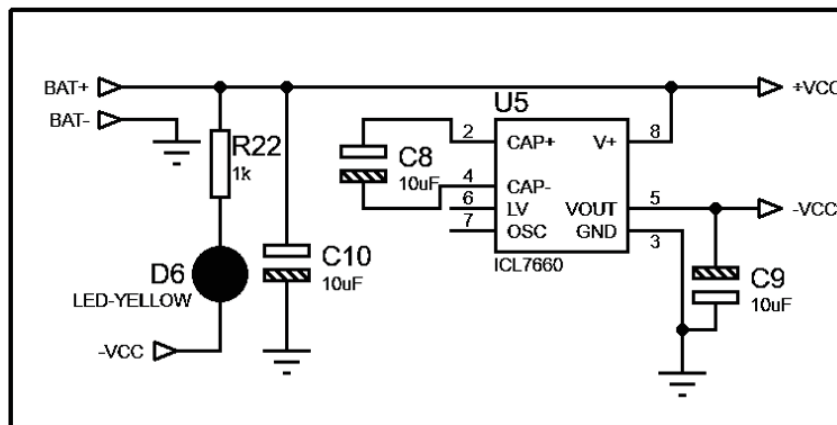
3.2 SENSOR ELETROMIOGRÁFICO

Para desenvolvimento do sensor eletromiográfico, utilizou-se um conjunto de circuitos combinados, sendo eles os de alimentação, de transdução, filtro passa-alta, filtro passa-baixa, amplificador digital, ajuste de off-set e o microcontrolador.

Como o intuito deste sistema eletromiográfico é eliminar fios e cabos, todo o circuito do

sensor foi alimentado por bateria, porém foi necessário a utilização de uma fonte simétrica, visto que o sinal de EMG possui informações em tensão positivas e negativas, dessa forma utilizou-se um circuito como fonte de alimentação que atenda essas necessidades por meio de um circuito integrado ICL7660 disposto na configuração conforme Figura 18. Nesta configuração, BAT+ e BAT- são os polos da bateria enquanto +VCC e -VCC são as saídas simétricas do circuito. A alimentação da bateria deve estar entre 3,3 V e 5 V.

Figura 18 – Circuito de alimentação



Fonte: Autoria própria (2021).

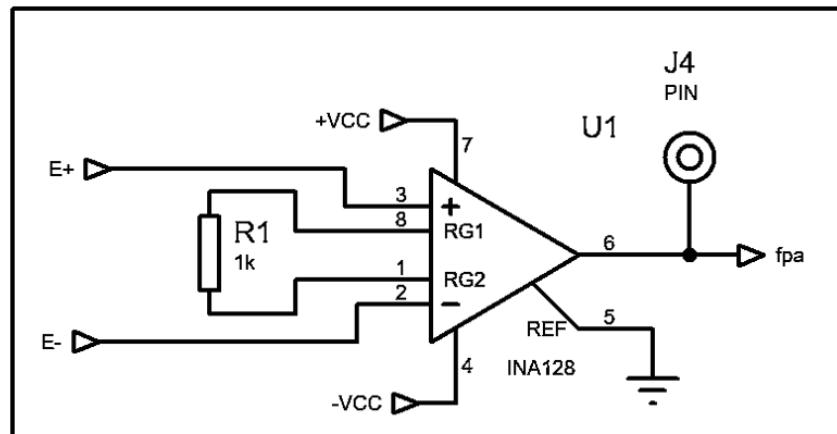
A aquisição do sinal ocorre através de um amplificador operacional diferencial de instrumentação, que realiza a transdução do sinal eletromiográfico. Para que esta etapa seja realizada adequadamente, é necessário que este componente possua alto nível de rejeição de modo comum (CMRR), acima de 90 dB, como recomendado por Merletti *et al.* (2008), com impedância de entrada superior a 10 mΩ, como sugerido por Konrad (2005), e com alta imunidade a ruídos. Para isso foi utilizado o INA128 do fabricante TEXAS INSTRUMENTS que atende estas principais características. No circuito, conforme Figura 19, R1 dita a amplificação do sinal diferencial em relação ao sinal comum as entradas, descrito pela equação 1.

$$G = 1 + 50K\Omega/R1 \quad (1)$$

As entradas E+ e E- são as entradas dos eletrodos de superfície e *fpa* é a saída do sinal para o próximo estágio do circuito, o filtro passa-alta.

As etapas de filtragem são de extrema importância para garantir que o espectro de frequência do sinal esteja limitado à faixa de interesse, eliminando os ruídos e sinais indesejados (VELLOSO, 2004). Segundo Bischoff *et al.* (1999), o potencial de ação possui componentes até 10 kHz de frequência dependendo da unidade motora, mas Becheli (2017) ressalta que a parte

Figura 19 – Circuito de transdução do sinal

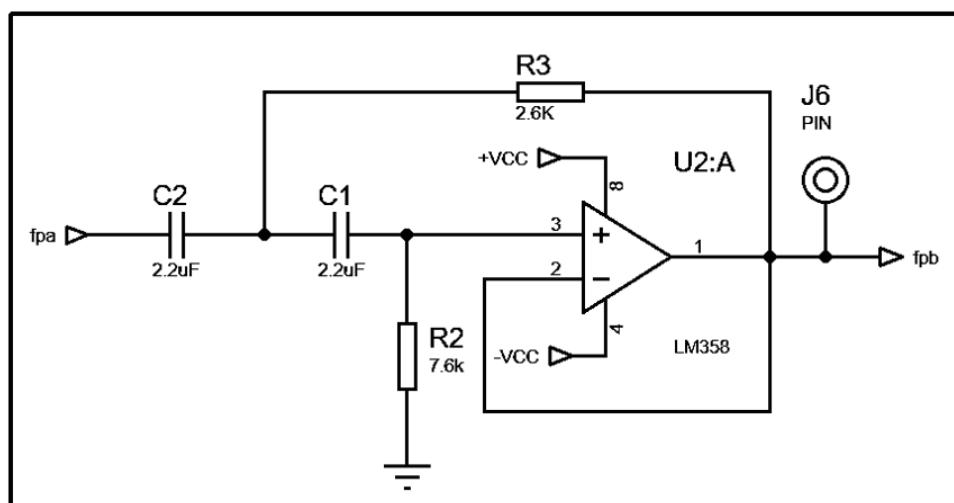


Fonte: Autoria própria (2021).

mais significativa da energia do sinal está concentrada entre 0 e 500 Hz. Dessa forma, Oliveira (2012) diz que é aceitável um filtro passa-alta com frequência de corte até 20 Hz, pois podem existir interferências de ondas de rádio e de artefatos de movimentos mecânicos nessa pequena faixa, enquanto o filtro passa-baixa deve possuir frequência de corte mínima de 500 Hz.

Os filtros foram desenvolvidos com o auxílio do *software* FilterPro da TEXAS INSTRUMENTS. Para a construção do filtro passa-alta foi utilizado um filtro do tipo Chebyshev 1 dB na configuração Sallen-Key de segunda ordem, com frequência de corte de 20 Hz, obtendo um fator de qualidade de 0,957 para componentes ideais, porém os componentes para esta especificidade não eram comerciais, dessa forma estes foram substituídos por outros com valores comerciais, obtendo assim o filtro apresentado na Figura 20.

Figura 20 – Circuito do filtro passa-alta

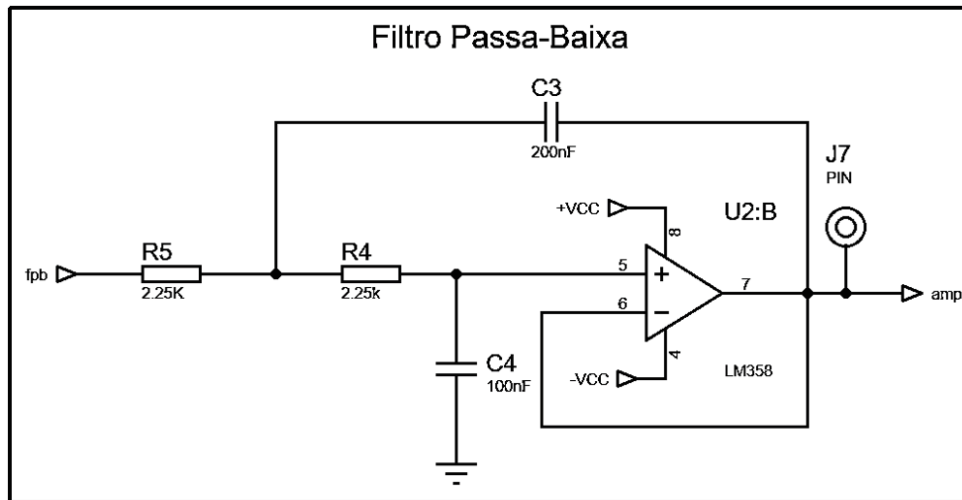


Fonte: Autoria própria (2021).

Para o filtro passa-baixa também foi utilizado um filtro do tipo Chebyshev 1 dB na

configuração Sallen-Key de segunda ordem, com frequência de corte de 500 Hz, obtendo assim um fator de qualidade também de 0,957 para componentes ideais. Assim, como no filtro passa-alta, os valores do componentes ideais tiveram que ser modificados para componentes comerciais, obtendo o filtro conforme Figura 21.

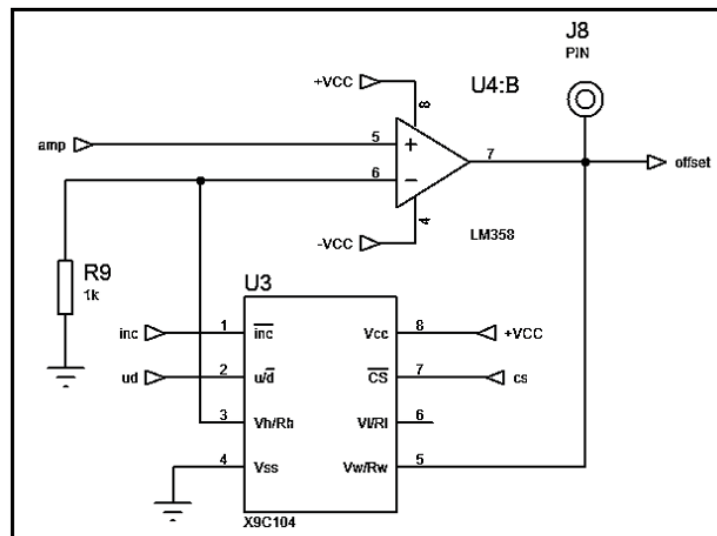
Figura 21 – Circuito do filtro passa-baixa



Fonte: Autoria própria (2021).

Após o sinal filtrado e foi amplificado, utilizou-se um circuito amplificador com controle digital baseado em uma configuração que utiliza um amplificador operacional e um resistor variável digital (Figura 22) o qual foi controlado pelo microcontrolador.

Figura 22 – Circuito do Amplificador Digital



Fonte: Autoria própria (2021).

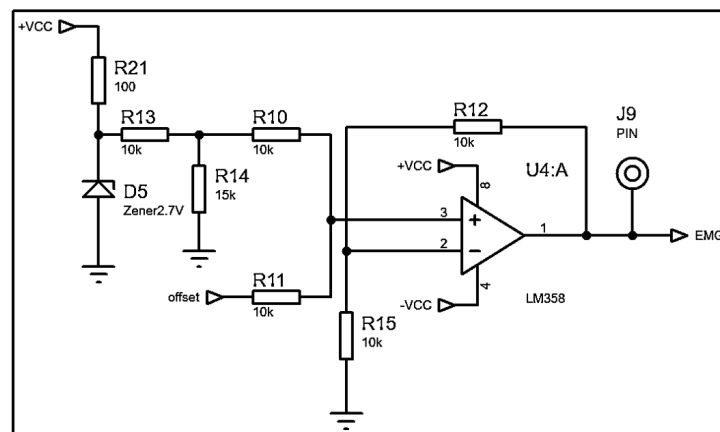
Este circuito amplificador possui ganho conforme a equação 2, ao qual R9 é um resistor de valor fixo e U3 é o circuito integrado X9C104, que pode variar sua resistência de $1K\Omega$ a

100K Ω através de controles digitais.

$$V_o = 1 + (R_{U3}/R_9) * V_{in} \quad (2)$$

Após a amplificação do sinal, é necessário um condicionamento do sinal eletromiográfico. Nesta etapa o sinal EMG possui componentes positivas e negativas, mas para permitir que microcontrolador realize as leituras do sinal de forma apropriada é necessário que possua apenas componentes positivas, para isso soma-se uma componente DC ao sinal, inserindo assim um nível de off-set. Isso permitirá que o microcontrolador realize leituras do sinal de forma apropriada. O circuito da Figura 23 será o responsável por realizar a soma da componente DC ao sinal de EMG.

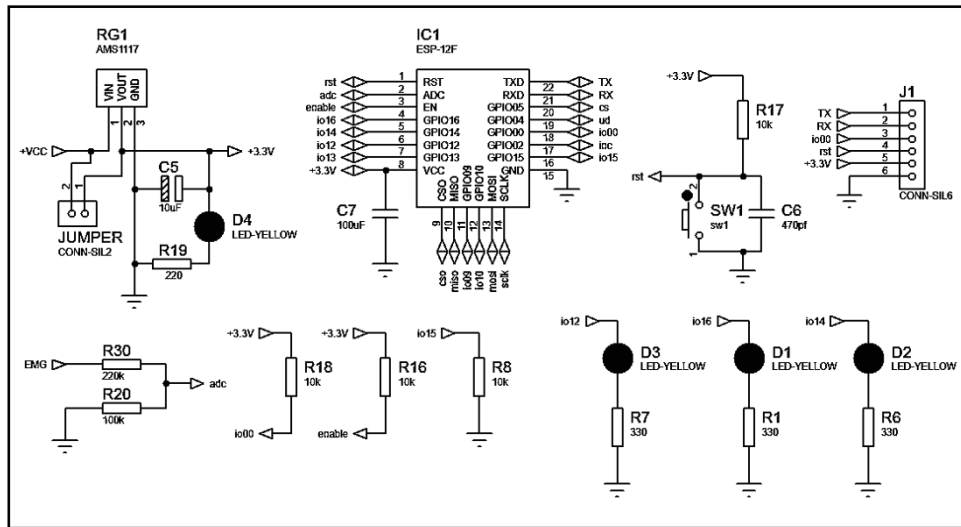
Figura 23 – Circuito de Off-set



Fonte: Autoria própria (2021).

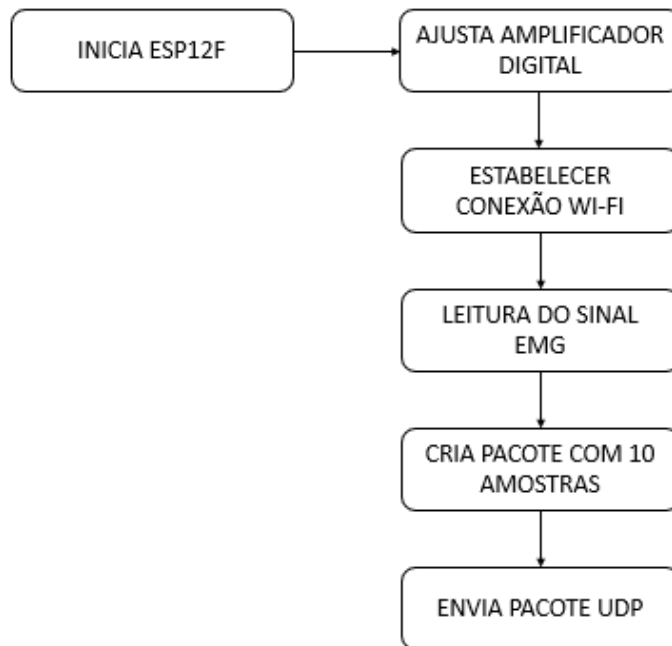
Após a realização de todo o tratamento analógico do sinal, o microcontrolador ficou encarregado de realizar a aquisição do sinal e transmiti-lo via Wi-Fi. Para melhorar a utilização e otimização de espaço optou-se pelo módulo ESP-12F, que possui como microcontrolador um ESP-8266, com porta de leitura analógica de 10 bits de resolução e antena de transmissão Wi-Fi integrada de 2.4GHz, operando a um *clock* de 80 MHz e com 4 MB de memória flash, atendendo as necessidades demandadas pelo projeto. Para amostragem do sinal eletromiográfico definiu-se um taxa de 1 kHz, que respeita a taxa de Nyquist, já que a frequência máxima de interesse do sinal foi limitada a 500 Hz pelo filtro passa-baixa. Utilizou-se ainda de LED's para aferição visual de identificação de envio de pacotes e de conexão Wi-Fi. Tais implementações foram dispostas conforme o circuito da Figura 24, com o firmware do microcontrolador executando conforme fluxograma da Figura 25.

Figura 24 – Circuito do microcontrolador



Fonte: Autoria própria (2021).

Figura 25 – Fluxograma do Firmware



Fonte: Autoria própria (2021).

3.3 INTERFACE

A interface teve por finalidade realizar a interação homem-máquina, a criação do servidor UDP, a recepção dos dados e o armazenamento deles.

Para a interação homem-máquina foi utilizado os recursos PyQrGraph e PyQt5 em linguagem de programação Python, pois segundo a Python Software FOUNDATION

PYTHON (2020), o PyQtGraph é um recurso de plotagem gráfica de alto desempenho feito unicamente em *Python* inserida em uma biblioteca GUI construída em PyQt4/PyQt5/PySide/PySide2 e *numpy*, que destina-se ao uso em aplicações matemáticas, científicas e de engenharia, para exibições 2D e 3D.

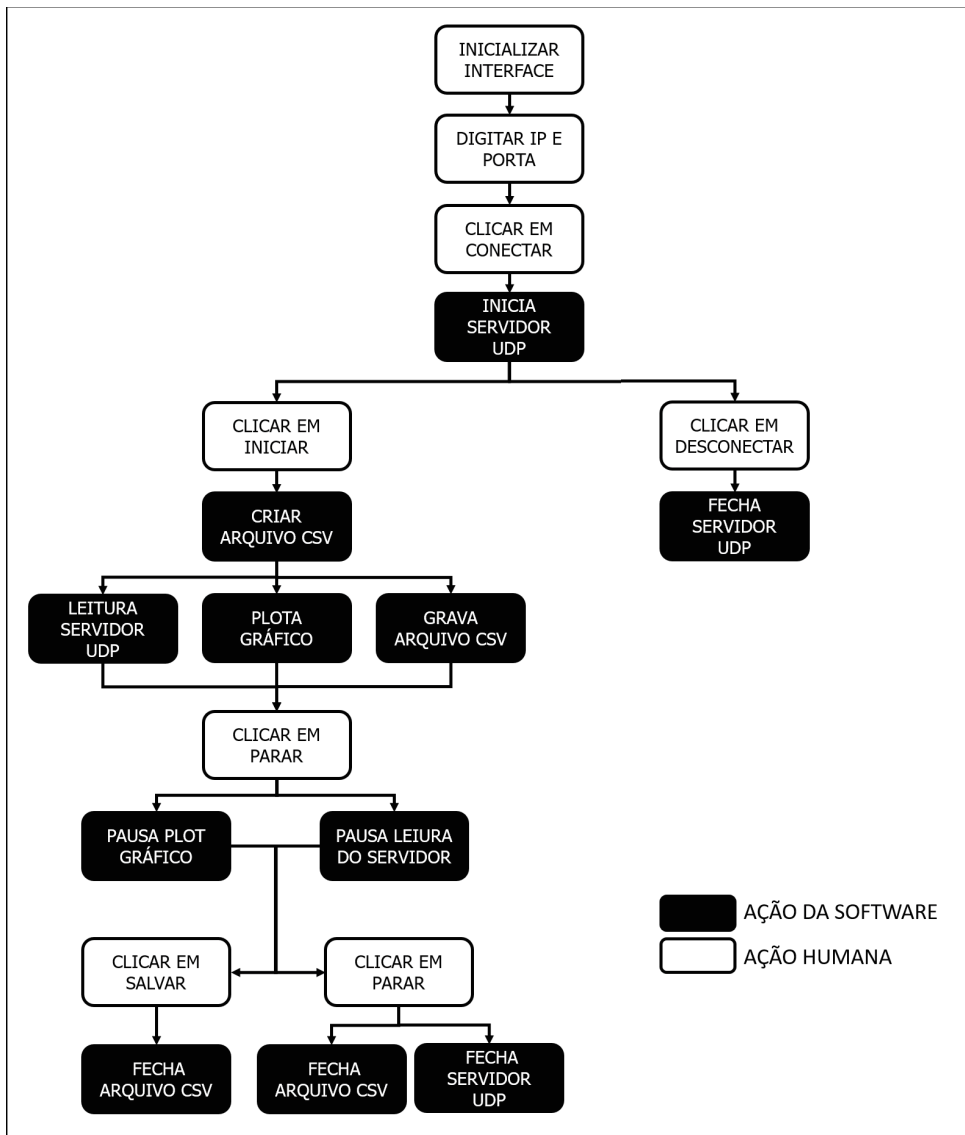
A criação e manutenção do servidor UDP foi realizado pela aplicação de interface através de uma biblioteca em Python, que ao ser iniciada executa o servidor UDP e passa a ler todas as informações enviadas a ele, e ao ser fechada, encerra a execução do servidor.

Para salvar todos os dados lidos do servidor, foi utilizado um arquivo no formato CSV (Comma Separated Values), já que este é o formato mais comum de importação e exportação de planilhas e bancos de dados. Segundo Python Software SOFTWARE FOUNDATION PYTHON (2021), o formato CSV ainda é muito utilizado, pois com a falta de um padrão bem definido, significa que existem diferenças entre os dados produzidos e consumidos por diferentes aplicativos, assim, enquanto os delimitadores e os caracteres de citação variam, o formato geral é semelhante o suficiente para que seja possível escrever um único módulo que possa manipular eficientemente esses dados, ocultando os detalhes da leitura e gravação dos dados do programador.

O fluxograma abaixo demonstra a execução de ações da interface e ações humanas.

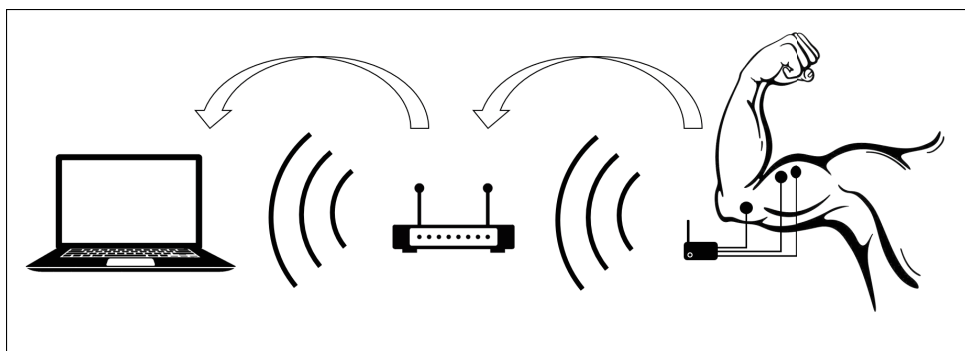
Com os processos da interface executando, tem-se um fluxo de dados, no qual o sensor eletromiográfico realiza as leituras dos impulsos elétricos musculares e os envia via rede Wi-Fi para um servidor UDP em um computador. A Figura 27 ilustra este processo. A interface realiza a leitura das informações do servidor e por meio de operações matemáticas, converte o valor digitalizado no valor real novamente, realizando então a gravação dos dados de valor real em um arquivo CSV e o plotando em gráfico na interface.

Figura 26 – Fluxograma de ações da interface



Fonte: Autoria própria (2021).

Figura 27 – Fluxo de Dados



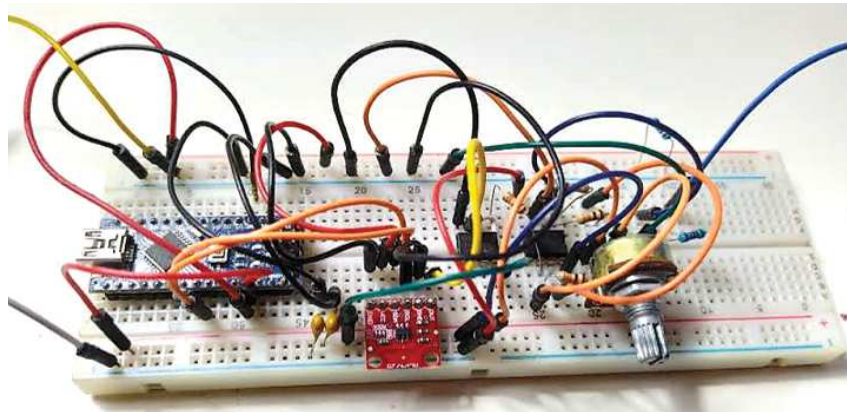
Fonte: Autoria própria (2021).

4 RESULTADOS

4.1 SIMULADOR ELETROMIOGRÁFICO

Para aferição do devido funcionamento do sistema simulador de sinal biológico (Figura 28), pontos específicos do circuito mostrado na Figura 17 foram utilizados como captura de sinal através de um osciloscópio, sendo eles na saída do circuito integrado MCP4725 (V_{out} - Pino 1 do MCP4725) e a saída do divisor de tensão (entre R7 e R8).

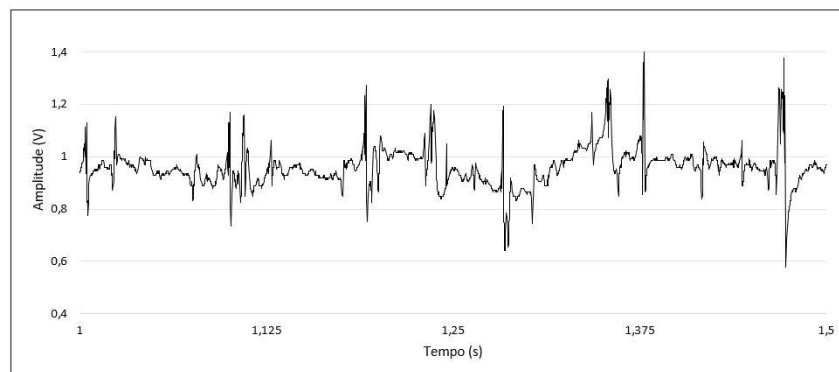
Figura 28 – Imagem do simulador de sinal biológico montado em protoboard



Fonte: Autoria própria (2021).

Para o primeiro ponto da saída do circuito integrado MCP4725, que tem como funcionalidade realizar a conversão digital-analógico do sinal transmitido pelo microcontrolador, obteve-se a Figura 29 para o sinal A (sinal obtido com eletrodo do tipo agulha).

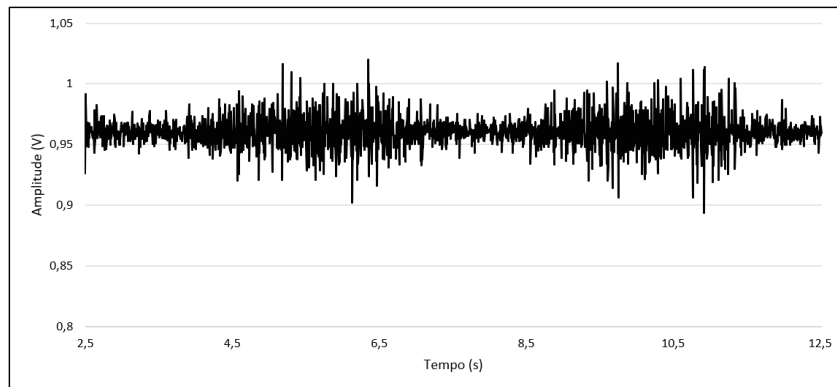
Figura 29 – Sinal eletromiográfico A - Saída do conversor digital-analógico



Fonte: Autoria própria (2021).

Já para o sinal B, obtido com o eletrodo de superfície, tem-se o sinal na saída do conversor digital-analógico, conforme Figura 30.

Figura 30 – Sinal eletromiográfico do eletrodo B - Saída do conversor digital-analógico



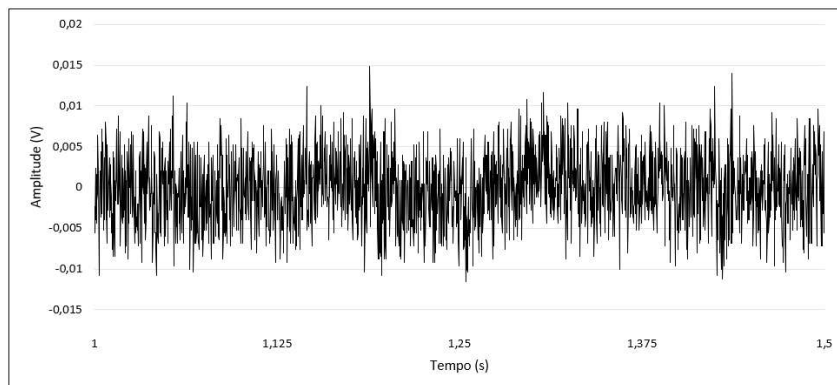
Fonte: Autoria própria (2021).

Percebe-se que os sinais da saída do conversor analógico-digital MCP4725 são similares aos sinais reais com seus condicionamentos de digitalização e adição de 1 volt como off-set, perdendo pouquíssima resolução em suas conversões, visto que ambos sinais já passaram por dois processos de transformação, analógico-digital e agora digital-analógico, sendo tendência perder precisão a cada transformação.

O segundo ponto que utilizou-se para obtenção de sinal do simulador foi a saída final do circuito, ou seja, no divisor de tensão. Para as componentes do sinal A obtendo o sinal conforme Figura 31, enquanto para o sinal B tem-se a Figura 32.

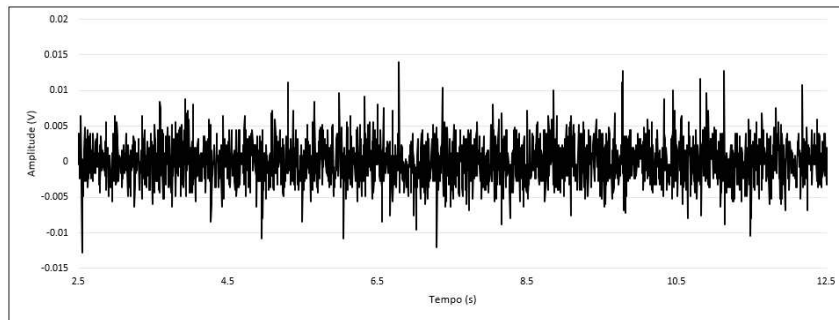
Este resultado é esperado uma vez que a medida que a amplitude do sinal é reduzida, a relação sinal ruído também é reduzida. Esta característica é importante para testar, nas próximas etapas, o correto funcionamento do circuito de aquisição e condicionamento do sinal analógico do EMG desenvolvido, que devem restaurar os sinais em meio ao ruído.

Figura 31 – Sinal eletromiográfico A - Saída do simulador



Fonte: Autoria própria (2021).

Figura 32 – Sinal eletromiográfico B - Saída do simulador

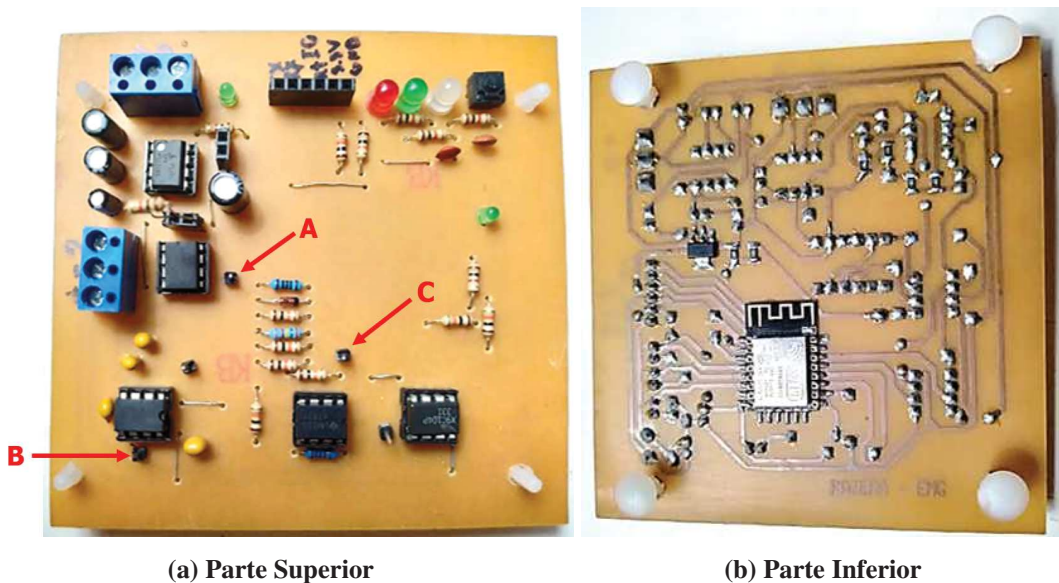


Fonte: Autoria própria (2021).

4.2 SENSOR ELETROMIOGRÁFICO

O sensor eletromiográfico desenvolvido, como mostrado pela Figura 33, utilizou-se da saída do circuito simulador de sinais biológicos como fonte de sinal para validação. Para isso conectou-se a saída do simulador aos dois pontos de entrada do amplificador operacional diferencial e uniu-se o terra (gnd) de ambos os circuitos.

Figura 33 – Sensor eletromiográfico desenvolvido



(a) Parte Superior

(b) Parte Inferior

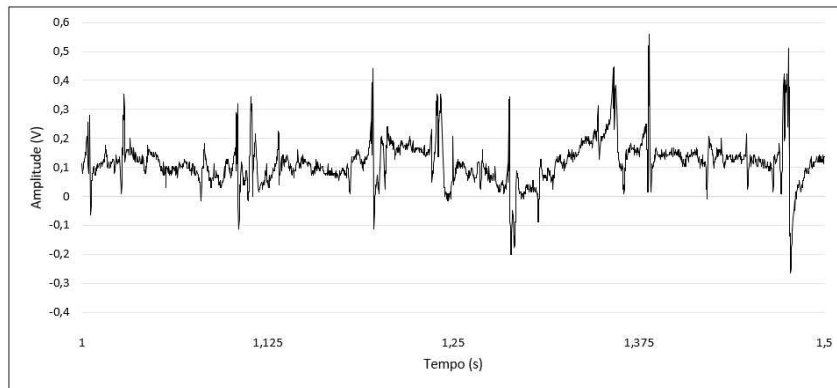
Fonte: Autoria própria (2021).

Para o sensor eletromiográfico utilizou-se vários pontos do circuito como coleta de sinal EMG, sendo a saída do amplificador operacional diferencial de instrumentação (A), na saída dos filtros (B) e na saída do circuito de ajuste de offset (C), conforme figura Figura 33(a), permitindo avaliar o sinal de entrada com pré amplificação, realizar a análise de desempenho dos filtros através do gráfico de bode e a saída do circuito de condicionamento antes do microcontrolador

realizar a leitura.

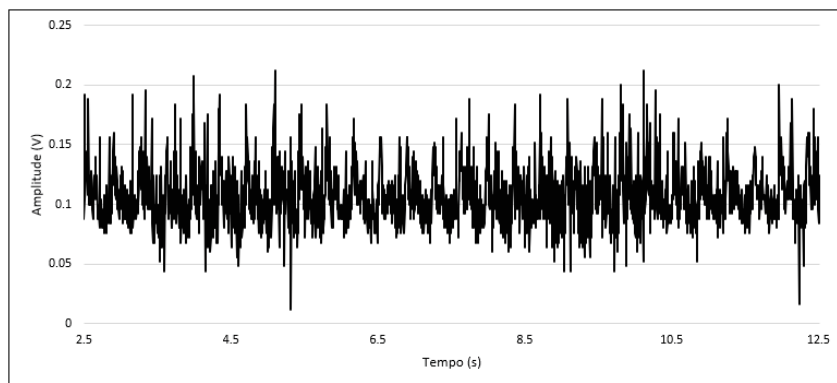
No primeiro ponto de aquisição de sinal de aferição, ou seja, na saída do amplificador operacional diferencial, no ponto J4 da figura Figura 19, para o sinal A obtém-se o sinal conforme Figura 34, enquanto para o sinal B tem-se a Figura 35.

Figura 34 – Sinal eletromiográfico A - Saída do Amplificador Operacional Diferencial



Fonte: Autoria própria (2021).

Figura 35 – Sinal eletromiográfico B - Saída do Amplificador Operacional Diferencial



Fonte: Autoria própria (2021).

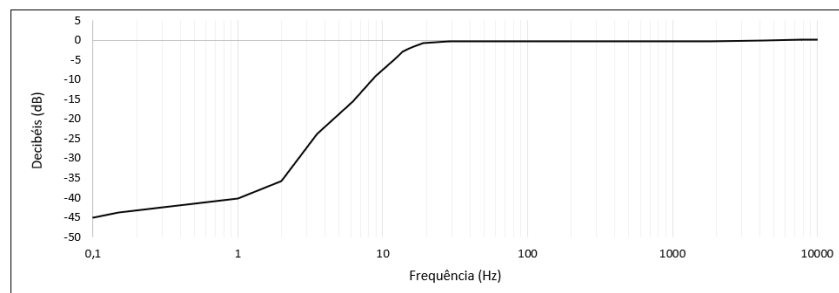
Percebe-se que o sinal A reconstituiu-se melhor do que o sinal B, eliminando os ruídos após a rejeição de modo comum (CMMR), como os ruídos são comuns em ambas as entradas do amplificador operacional diferencial ($E+$ e $E-$), os ruídos tendem a ser atenuados em relação ao sinal, dados pelo resistor R1, resultando em 51 vezes de diferença a relação de sinal/ruído, e essa taxa pode chegar até 110 dB, conforme indicado pelo fabricante de circuito integrado INA 128 (TEXAS INSTRUMENTS, 2005).

Na etapa de filtros, utilizou-se como desempenho o nível de atenuação através dos gráficos de Bode obtidos de cada filtro. Segundo Konrad (2005), quase toda a potência de um sinal EMG está localizada entre 10 e 250 Hz, mas a aquisição do sinal requer uma configuração

de banda do amplificador de 10 a 500 Hz. Dessa forma, foram implementados um passa-baixas com frequência de corte de 500Hz e um passa-altas com corte em 20 Hz, dispostos em série, formando um passa-faixa.

O filtro passa-altas (Figura 20) atuará em todas as frequências abaixo de 20Hz, porém alguns componentes não estavam no padrão comercial, dessa forma utilizou-se dos componentes com valores mais próximos. Isso fez com que nossa curva projetada fosse deslocada, mas não o suficiente para interferir no funcionamento do sistema. É possível visualizar na Figura 36 como esse deslocamento se deu, fazendo a frequência de corte, que é dada em 3dB de atenuação, sair de 20Hz para próximo dos 15Hz.

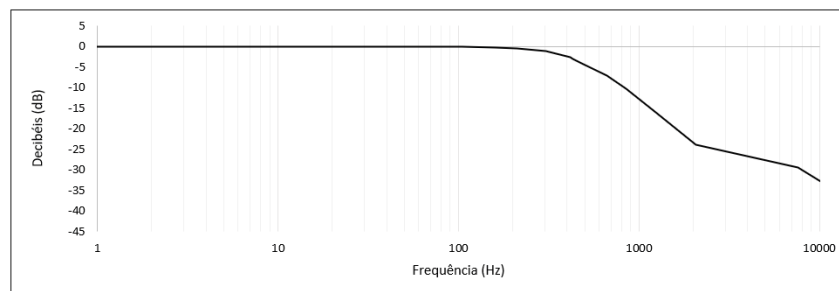
Figura 36 – Gráfico de bode do filtro passa-altas



Fonte: Autoria própria (2021).

No passa-baixas (Figura 21), ocorreu a mesma situação do filtro passa-altas, os valores comerciais dos componentes influenciaram no comportamento dos filtros, fazendo com a curva de corte fosse modificada. Nesse caso, o filtro implementado para 500Hz de frequência de corte, teve seu corte próximo a 470Hz, perdendo uma pequena faixa de sinal de interesse, porém essa faixa perdida possui a menor potência dentre toda a faixa requerida. Na Figura 37 é apresentada a resposta em frequência do filtro passa-baixas.

Figura 37 – Gráfico de bode do filtro passa-baixas

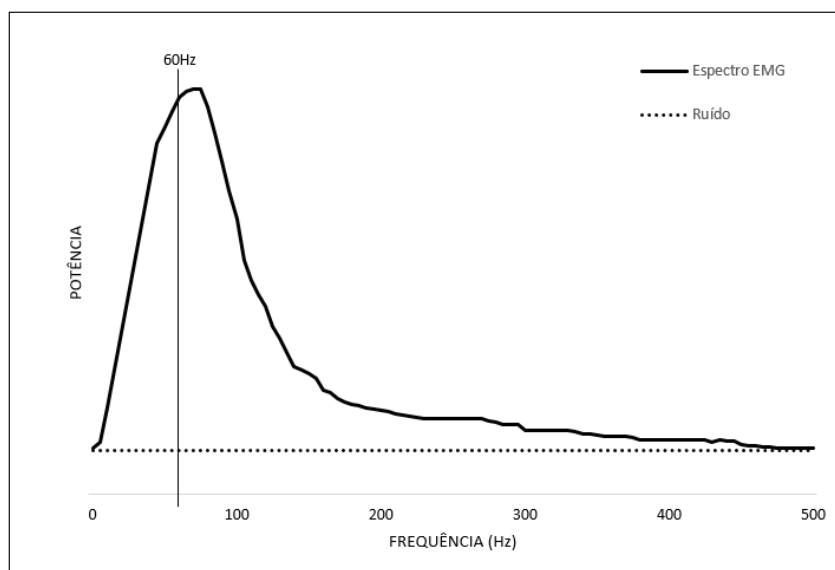


Fonte: Autoria própria (2021).

Um filtro *notch* em 60Hz geralmente é implementado nesse tipo de sistema para atenuar ruídos provindos principalmente da rede elétrica, mas Bischoff *et al.* (1999) diz que esse tipo

de filtro pode esconder componentes interessantes do sinal e até mesmo modificar a fase do sinal abruptamente para frente e para trás em torno do entalhe da frequência, podendo assim distorcer a forma de onda. Dessa forma, durante o processo de condicionamento do sinal não implementou-se um filtro *notch*, pois mesmo podendo ocorrer interferências em 60Hz, existe eletromiográfico nesta faixa. Na Figura 38 é mostrado a distribuição do espectro de potência comumente ocorrida em um sinal EMG. Assim Hermens *et al.* (2012) recomenda que qualquer filtragem Notch, para cancelar algum sinal dentro da faixa de interesse EMG deve ser evitada, pois destrói muitas informações de sinal.

Figura 38 – Espectro de sinal eletromiográfico



Fonte: Adaptado de Hewson *et al.* (2003)

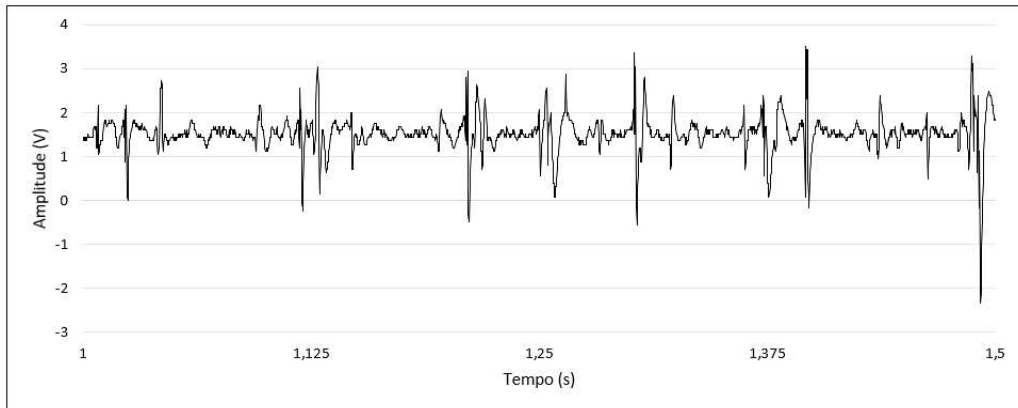
Na etapa de condicionamento final do sinal tem-se os circuitos de amplificação (Figura 22) e inserção de offset (Figura 23). A amplificação é realizada pelo circuito integrado X9C104, que conforme projeto do eletromiógrafo permite amplificação de 0 a 100 vezes, sendo ajustada através do *firmware* contido no ESP12F, durante os testes foram utilizados os valores em *firmware* de 4 a 10 como taxa de amplificação. Considerando a etapa de pré amplificação no circuito integrado INA128, que estava por amplificar em 51 vezes o sinal, o X9C104 foi ajustado para 5 vezes, totalizando um amplificação total de 255 do sinal provindo do simulador.

O circuito de off-set disposto pós amplificador, tem por função adicionar 1,5V ao sinal EMG. Visto que o EMG pode conter amplitude de até 10 mV, o mesmo amplificado em 255 vezes, chega a ter amplitude máxima de 2,55 V; porém, o mesmo é um sinal alternado que distribui os 2,55 V na faixa de 1,3 V a -1,3 V. Assim, ao adicionar 1,5V ao sinal eletromiográfico, este passa a ter sua variação dentro da faixa de 0,2 V a 2,75 V, permitindo ao microcontrolador

realizar a leitura analógica do sinal sem que ocorra sua saturação durante essas duas etapas de condicionamento.

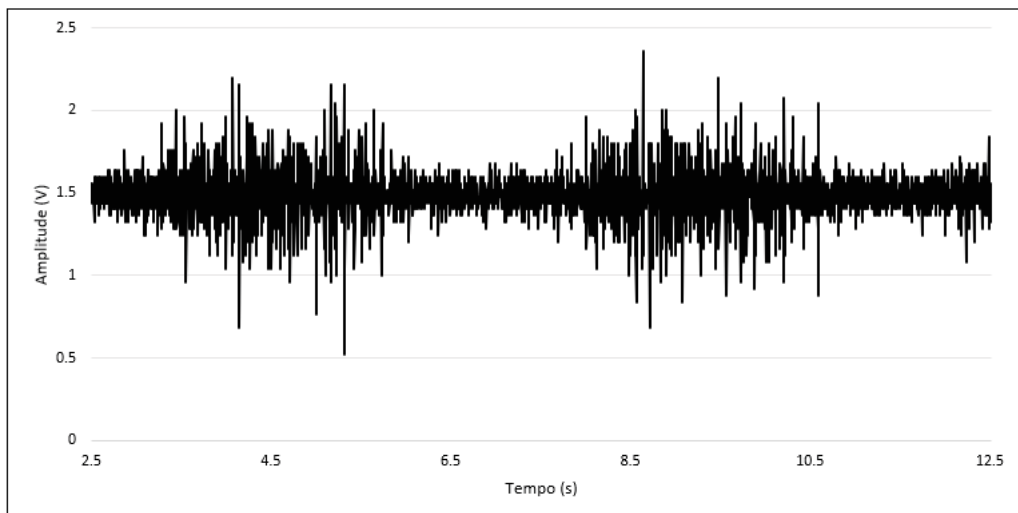
Na saída do circuito de condicionamento, tem-se o mesmo sinal ao qual o microcontrolador realizará a leitura. Utilizando-se os sinais simulados A e B, através de um osciloscópio obtém-se as formas de onda conforme a Figura 39 e Figura 40.

Figura 39 – Sinal A condicionado



Fonte: Autoria própria (2021).

Figura 40 – Sinal B condicionado



Fonte: Autoria própria (2021).

A forma de onda do sinal A, sendo este um sinal bem detalhado, manteve sua consistência em seus picos de maior amplitude; porém, teve certa distorção em suas partes que possuíam menor oscilação, ou seja, durante todo o processo de condicionamento a parte oscilatória do sinal tendeu a se manter consistente enquanto os trechos com menor oscilação tendeu a se tornar uma oscilação ao nível imposto de off-set. Além disso, na Figura 39 também nota-se que o nível de amplificação utilizado foi 7 no circuito do X9C104, o que resultará em uma provável saturação

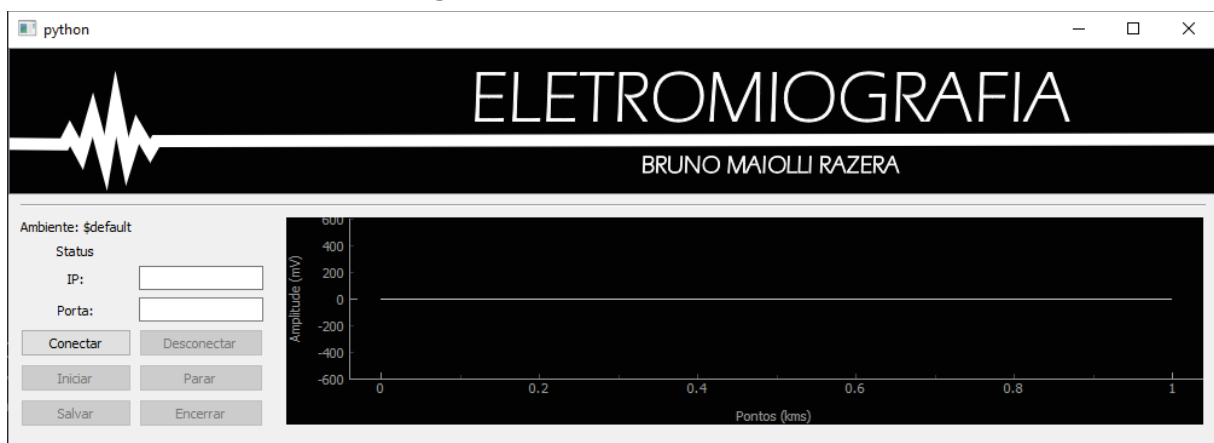
dos picos durante a leitura do microcontrolador, já que este consegue realizar leituras de 0V a 3.3V apenas.

Na Figura 40 tem-se a leitura do sinal B, na qual está mais próximo do sinal real ao qual o circuito realizará leituras, que manteve-se consistente em seu conjunto de componentes, mantendo-se quase fiel ao sinal original. Este utilizou um fator de 4 para o circuito integrado X9C104, que resultou em um bom nível de amplificação sem atingir a saturação do sinal.

4.3 INTERFACE

Para a interação homem-máquina do sistema eletromiográfico tentou-se criar uma interface que fosse amigável e de utilização fácil e sugestiva, como pode-se ver na Figura 41.

Figura 41 – Tela inicial da Interface



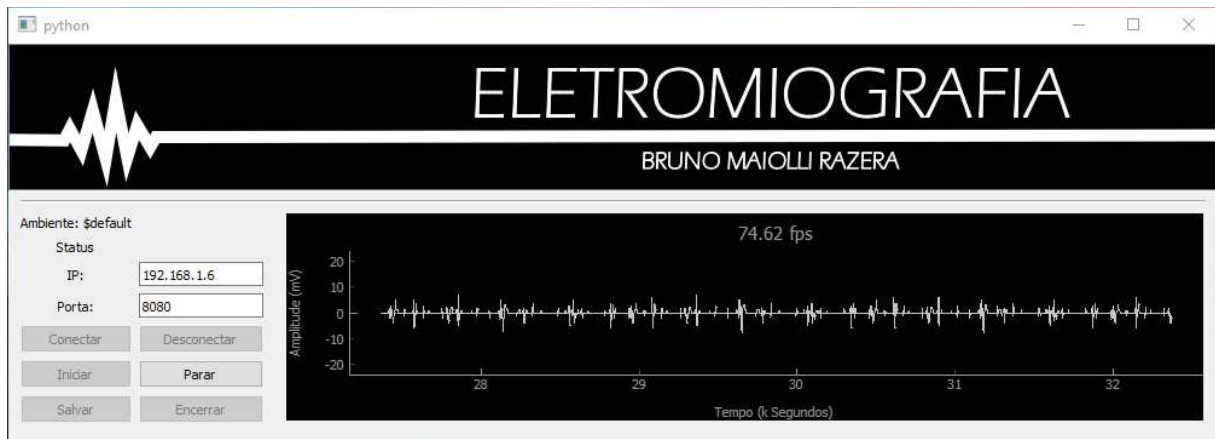
Fonte: A autoria própria (2021).

Para a utilização do sistema o usuário deve apenas digitar o endereço IP (*Internet Protocol*) e a porta de acesso da máquina ao qual a interface está sendo executada, podendo ser um notebook, uma computador desktop ou até mesmo um em hardware embarcado e clicar em conectar, que o software criará um servidor UDP para transição de dados e realizará automaticamente as ações de leitura do sinal EMG transmitido, plotagem gráfica na interface e salvamento dos dados lidos.

A plotagem gráfica dos sinais utilizados como validação na interface apontou deformação visual nos gráficos, de forma que ao comparar com o gráfico original, o sinal do gráfico da interface apresentará maior estendimento no eixo X, já que os gráficos possuem dimensões diferentes, mas o sinal continua coerente com o original em ambos os casos. Para o sinal invasivo, obtém-se uma perda visual nas partes de pequena amplitude do sinal, devido a compressão

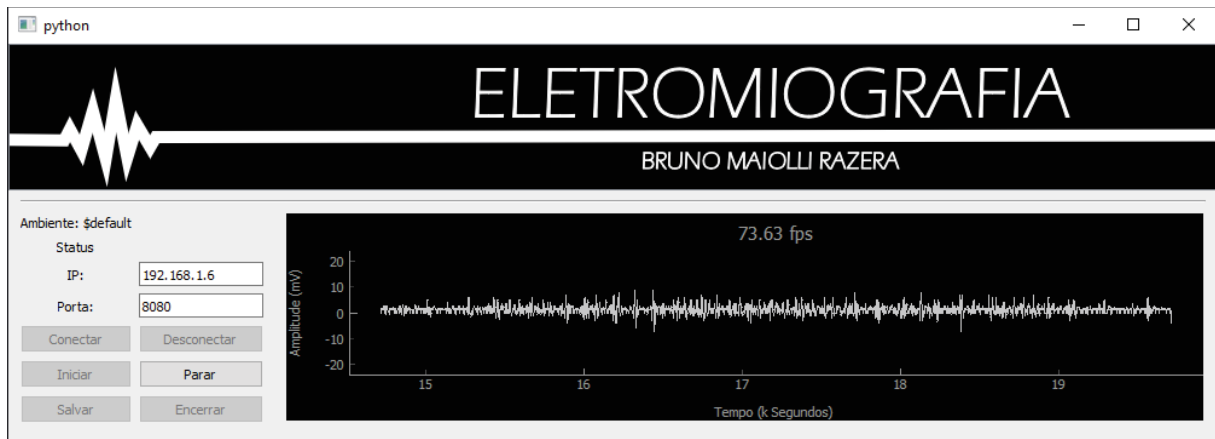
no eixo Y, sendo possível visualizar apenas os picos do sinal, como pode-se ver na Figura 42, enquanto para o sinal não invasivo consegue-se ter melhor visualização, visto que este em geral possui maior amplitude e uma maior quantidade de picos, como visualiza-se na Figura 43. Como o sinal não invasivo é próximo ao sinal real obtidos pelos eletrodos do sistema, um sinal em condições reais terá melhor representação na interface gráfica.

Figura 42 – Interface em funcionamento - plotagem do sinal A



Fonte: Autoria própria (2021).

Figura 43 – Interface em funcionamento - plotagem do sinal B

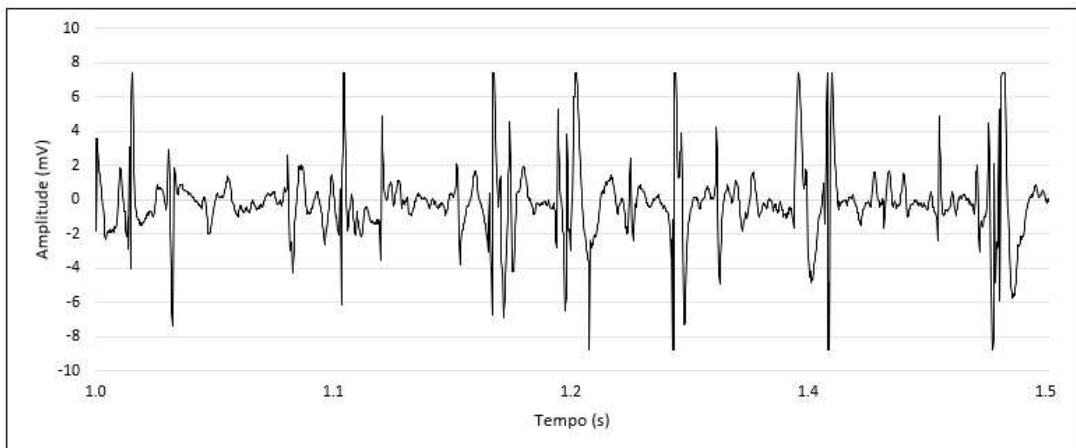


Fonte: Autoria própria (2021).

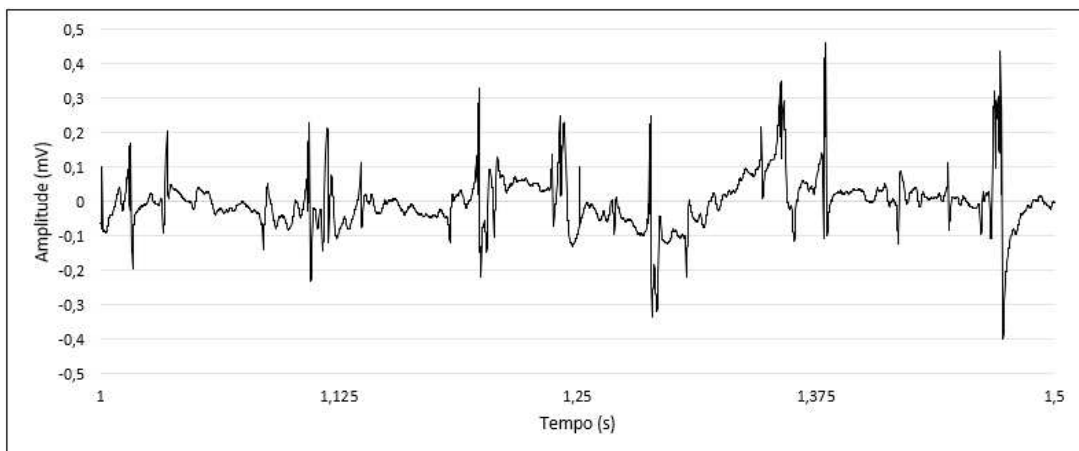
Durante o processo de plotagem gráfica, para cada pacote de dados realizado a leitura do servidor, ocorrerá a gravação do mesmo em um arquivo CSV, na qual terá o nome do arquivo e na primeira linha do arquivo o cabeçalho do sensor que está enviando sinal ao servidor e em cada linha será registrado um ponto de leitura do gráfico, dessa forma, como a taxa de amostragem do sistema respeita a taxa de Nyquist, sendo de 1KHz, a cada mil linhas tem-se a representação de um segundo de leitura real, sendo possível reconstituir graficamente os sinais do sistema de eletromiografia.

Na Figura 44 tem-se a saída dos dados salvos em CSV e do sinal original, e assim como discutido anteriormente, ocorreram perdas no sinal de forma a aumentar a amplitude dos picos e minimizar a amplitude de trechos com menores oscilações, mas ainda mantendo-se próximo a forma original do sinal. Já para o sinal de superfície visualiza-se que este manteve próximo ao original, conforme Figura 45 visto que este possui maiores oscilações ao decorrer do sinal coletados e ser um sinal em escala de tempo maior, minimizando assim a visualização de pequenos sinais ao decorrer do gráfico. Outro ponto de possível análise é a consistência do sinal ao decorrer do tempo, como pode ser visto na Figura 46, que ao realizar a plotagem de 20 segundos do sinal de superfície repetidamente, o mesmo manteve próximo as suas características originais em todas as repetições e entre elas mesmas, demonstrando pouquíssima variação do sistema eletromiográfico ao longo do tempo.

Figura 44 – Comparativo do Sinal eletromiográfico A

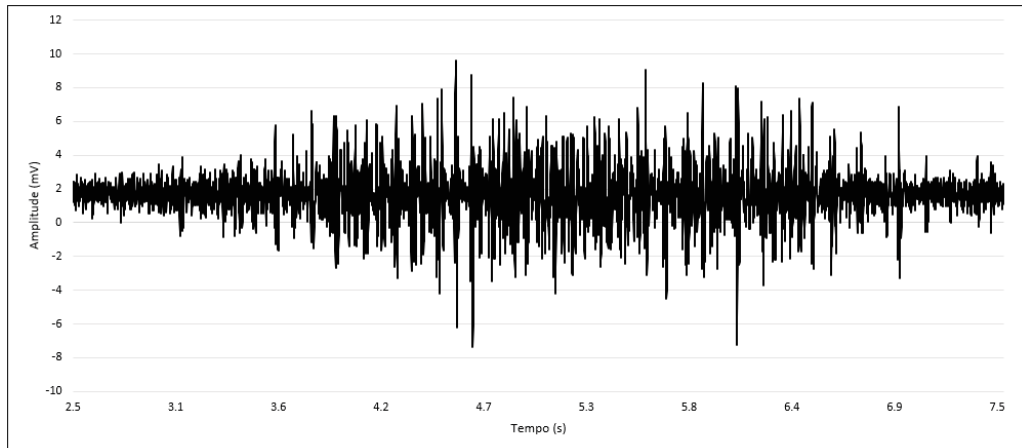
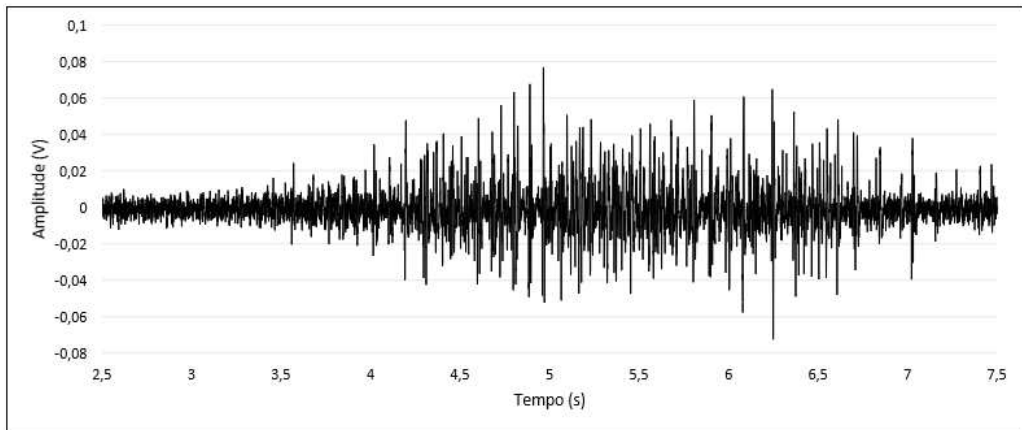


(a) Dados salvo em CSV

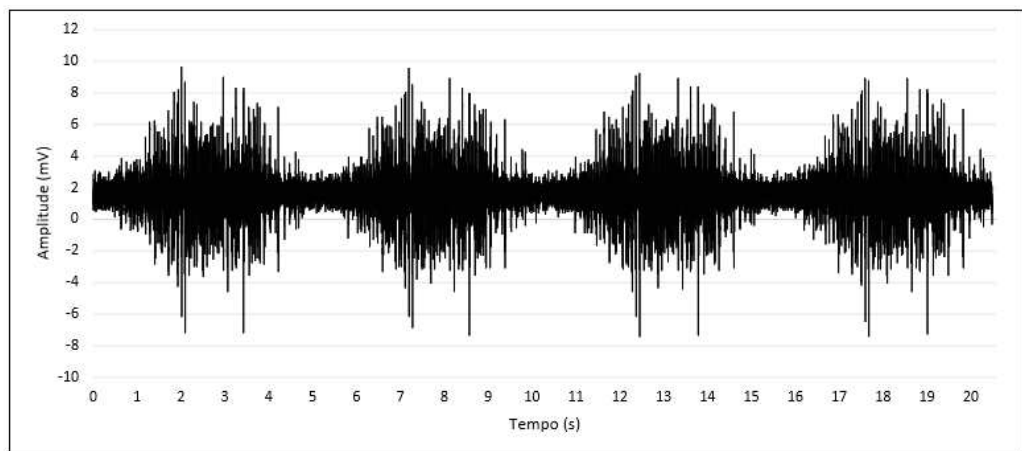


(b) Sinal Original

Fonte: Autoria própria (2021).

Figura 45 – Comparativo do sinal eletromiográfico B**(a) Dados salvo em CSV****(b) Sinal Original**

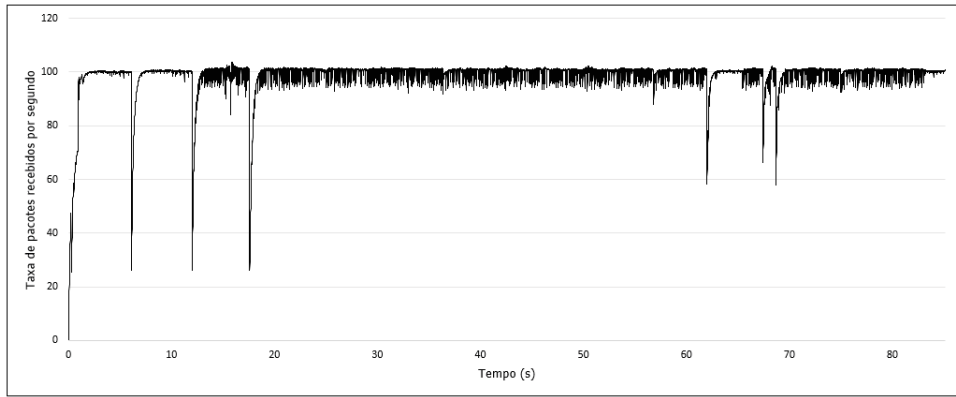
Fonte: Autoria própria (2021).

Figura 46 – Sinal eletromiográfico B - Dados salvo em CSV

Fonte: Autoria própria (2021).

O sistema eletromiográfico projetado visa operar em tempo real, ou seja, com o menor tempo de atraso possível durante sua transmissão. Durante testes realizados de forma empírica, constatou-se que a taxa de informações do servidor era de 400 pacotes por segundo (dependendo do *hardware*), enquanto a atualização máxima do gráfico era de 200 atualizações por segundo e como a taxa de leitura do sinal é de 1000 Hz, tornou-se necessário realizar ajustes para a correta transmissão. O método encontrado para uma transmissão foi realizar a transmissão de várias leituras por pacote, com o microcontrolador formando *buffer* de 10 leituras, e transmitindo para o servidor como um único pacote, fazendo com que a taxa de transmissão seja de 100 Hz, e o gráfico passou a ter taxa de atualização de 100 vezes por segundo, atualizando 10 pontos por vez.

Durante a transmissão de pacotes, o tempo entre a chegada dos pacotes na interface passaram a ser monitorados, de forma a se obter uma taxa de pacotes por segundos, que são representados conforme Figura 47. Visualizou-se que as oscilações na rede são muito perceptíveis pelo sistema, principalmente por se tratar de uma rede local caseira com diversos dispositivos conectados na mesma, acabando por gerar picos de atrasos na transmissão e/ou perdas dos pacotes. Quando a rede tende a estabilidade, os pacotes com atraso acabam por chegar ao servidor com um tempo menor que a taxa de transmissão realizada em relação ao último pacote enviado, ou seja, com um menor tempo entre os pacotes, maior será a taxa de transmissão, de forma a ultrapassar rapidamente os 100 pacotes por segundos e logo em seguida retornando a taxa esperada. Esse processo de oscilação ocorre minimamente durante todo o período de transmissão podendo ser geradas por atrasos de processamento (tempo para examinar o cabeçalho do pacote e determinar para onde o mesmo deverá ser encaminhado), atrasos de fila (tempo que cada pacote espera em fila nos roteadores antes de ser encaminhado ao próximo enlace), atrasos de transmissão (tempo necessário para que cada bit do pacote seja empurrado para dentro do enlace) e o atraso de propagação (tempo de viagem do pacote até o próximo nó), sendo estes cumulativos (KUROSE; ROSS, 2010).

Figura 47 – Taxa de Pacotes recebidos por segundos

Fonte: Autoria própria (2021).

5 CONCLUSÃO

O objetivo do trabalho foi desenvolver um sistema de monitoramento eletromiográfico de superfície com comunicação sem fio, transmitindo os dados monitorados do grupo muscular para uma interface gráfica em tempo real com opção de armazenamento dos dados recebidos, que foram atingidos conforme esperados.

O protótipo desenvolvido pode ser utilizado em situações reais cumprindo satisfatoriamente sua proposta, apresentando boa resposta e integração dos componentes do sistema, desde a captação de dados até a apresentação de forma gráfica e armazenamento desses dados. Além disso, o mesmo possui a possibilidade de reduzir significativamente suas dimensões utilizando componentes SMD no lugar de PTH, melhorando sua função como sistema portátil.

A utilização do conjunto, uso bateria e comunicação sem fio, facilita o monitoramento dos grupos musculares, pois não existe as limitações de movimentos impostas por fios e aumenta a distância de exame em relação ao aparelho de plotagem gráfica de dados. Estes fatores ainda permitem a capacidade de escalabilidade do sistema, ou seja, utilizar vários sensores ao mesmo tempo, pois sua transmissão sem fio facilita a integração de vários sensores em rede, sendo necessário apenas alguns ajustes na interface gráfica. Além disso, o uso bateria reduz os ruídos que poderiam ser providos dos movimentos dos fios e da rede elétrica.

A transmissão de pacotes contendo 10 amostras, se mostrou eficiente em relação a proposta, possuindo assim um pequeno atraso na transmissão do pacote, gerando um atraso visualmente imperceptível na interface, possibilitando a análise do movimento, conforme o mesmo é executado. Dessa forma, pode-se considerar este sistema como um sistema de tempo real.

A continuação desse projeto pretende agregar melhorias vistas ao decorrer da obtenção de resultados, tais como implementar uma comunicação bidirecional da interface com o sensor eletromiográfico, podendo assim assimilar todos os comandos através da interface; melhorar o design da plotagem gráfica; implementar suporte multicanal ao sistema para utilização de vários sensores simultaneamente; comparar o sistema desenvolvido neste projeto com aparelhos clínicos com validação e realizar ensaios para verificar se o sensor atende as normas ABNT NBR IEC 60601-2: Requisitos particulares para a segurança básica e o desempenho essencial de eletromiógrafos.

REFERÊNCIAS

- ALASKER, A. M. Electromyography control of a computer model of the arm. **Journal of Biomedical Science and Engineering**, Scientific Research, v. 07, n. 13, 2014.
- BECHELI, M. H. **Módulo de Eletromiografia**. 2017. Monografia (Bacharel em Engenharia Elétrica), Universidade Estadual de Londrina, Londrina, Brasil.
- BISCHOFF, C.; FUGLSANG-FREDRIKSEN, A.; VENDELBO, L.; SUMNER, A. Standards of instrumentation of emg. **Clinical Biomechanics**, 1999.
- BORGES, A. P. **Desenvolvimento de um sistema holter de eletrocardiograma**. 2018. Monografia (Bacharel em Engenharia Eletrônica), Universidade Tecnológica Federal do Paraná, Campo Mourão, Brasil.
- BOYLESTAD, R. L. **Introdução à análise de circuitos**. 12. ed. São Paulo: Pearson Prentice Hall, 2012.
- CHAUVET, E.; FOKAPU, O.; HOGREL, J. Y.; GAMET, D.; DUCHÊNE, J. Automatic identification of motor unit action potential trains from electromyographic signals using fuzzy techniques. **Medical and Engineering and Computing**, v. 41, n. 6, 2003.
- FORTI, F. **Análise do sinal eletromiográfico em diferentes posicionamentos, tipos de eletrodos, ângulos articulares e intensidade de contração**. 2005. 146 p. Dissertação (Pós-Graduação em Fisioterapia) — Universidade Metodista de Piracicaba, Piracicaba, 2005.
- GARCIA, M. A. C.; MAGALHÃES, J.; IMBIRIBA, L. A. Comportamento temporal da velocidade de condução de potenciais de ação de unidades motoras sob condições de fadiga muscular. **Revista Brasileira de Medicina do Esporte**, v. 10, n. 4, 2004.
- GUGELMIN, F. **Internet: qual a diferença entre os protocolos UDP e TCP?** 2014. Disponível em: <https://www.tecmundo.com.br/internet/57947-internet-diferenca-entre-protocolos-udp-tcp.htm>. Acesso em: 01 dez. 2020.
- HERMENS, H. J.; FRERIKS, B.; MERLETTI, R.; STEGEMAN, D.; BLOK, J.; RAU, G.; DISSELHORST-KLUG, C.; HAGG, G. European recommendations for surface electromyography. **SENIAM project**, Roessingh Research and Development, 2012.
- HEWSON, D. J.; HOGREL, Y. J.; LANGERON, Y. Evolution in impedance at the electrode-skin interface of two types of surface emg electrodes during long-term recordings. *Journal*

of electromyography and kinesiology : official journal of the International Society of Electrophysiological Kinesiology, v. 13, p. 273–279, 2003.

HUHTA, J. C.; WEBSTER, J. G. 60-hz interference in electrocardiography. **IEEE Transactions on biomedical engineering**, v. 20, n. 2, 1973.

IDE, B. N.; RAMARI, C.; MURAMATSU, L. V.; VIEIRA, W. F.; BIANCHI, S.; RISSI, R.; MACEDO, D. V.; PALOMARI, E. T. Eletromiografia de superfície - aplicações na fisiologia do exercício. **Revista Acta Brasileira do Movimento Humano**, v. 2, n. 4, p. 60–78, 2012.

IEE-LAB. **IEE EMG DATABASE**. 2017. Disponível em: https://www.ufrgs.br/ieelab/IEE_sEMG_db.php. Acesso em: 10 nov. 2020.

KONRAD, P. **The ABC of EMG: A practical introduction to kinesiological electromyography**. 1. ed. Scottsdale: NORAXON INC. USA., 2005.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: Uma abordagem top-down**. São Paulo: Pearson, 2010.

LADEGAARD, J. **Story of electromyography equipment: Muscle and nerve**. [S.l.]: A Wiley Company, 2010.

LUCA, C. J. Surface electromyography: Detection and recording. **DelSys Incorporated**, 2002.

LUCA, C. J. Lectromyography. **Encyclopedia of Medical Devices and Instrumentation**, Boston, EUA, p. 98–109, 2006.

MAGALHÃES, L. **Bomba de Sódio e Potássio**. 2017. Disponível em: <https://www.todamateria.com.br/bomba-de-sodio-e-potassio/>. Acesso em: 01 nov. 2019.

MALMIVUO, J.; PLONSEY, R. **Bioelectromagnetism: Principles and applications of bioelectric and biomagnetic fields**. 1. ed. Nova York: Oxford University Press, 1995.

MALVINO, A.; BATES, D. J. **Eletrônica**. AMGH, Porto Alegre, v. 2, 2016.

MERLETTI, R.; BOTTER, A.; TROIANO, A.; MERLO, E.; MINETTO, M. A. Technology and instrumentation for detection and conditioning of the surface electromyographic signal: State of the art. **Clinical Biomechanics**, v. 28, n. 24, p. 122–134, 2008.

MICROCHIP. **12-Bit Digital-to-Analog Converter with EEPROM Memory in SOT-23-6**. [S.l.], 2007. Disponível em: <https://ww1.microchip.com/downloads/en/DeviceDoc/22039d.pdf>. Acesso em: 10 jan. de 2021.

NARAYAN, A.; DASGUPTA, H.; PAL, H. Instrumentation of a portable emg sensor in a single pcb for human motion pattern detection. **Symposium on Physics and Technology of Sensors**, Pune, India, n. 2, 2015.

NODA, D. K. G.; MARCHETTI, P.; VILELA JUNIOR, G. A eletromiografia de superfície em estudos relativos à produção de força. **Revista CPAQV – Centro de Pesquisas Avançadas em Qualidade de Vida**, v. 6, n. 3, 2014.

OLIVEIRA, F. A. **Sensor de Eletromiografia sem fio a seco com acelerômetro triaxial integrado**. 2012. 96 p. Dissertação (Programa de Pós-Graduação em Engenharia Elétrica) — Universidade Federal de Itajuba, Itajubá, 2012.

ORESTES, Y. **TCP, UDP e protocolos: quais as diferenças?** 2019. Disponível em: <https://www.alura.com.br/artigos/quais-as-diferencas-entre-o-tcp-e-o-udp>. Acesso em: 02 dez. 2020.

PHYSIONET. **Examples of Electromyograms**. 2016. Disponível em: <https://archive.physionet.org/physiobank/database/emgdb/>. Acesso em: 06 jul. 2020.

RODRIGUEZ-AÑEZ, C. R. A eletromiografia na análise da postura. 2000.

SILVEIRA, D. P. **O que é o modelo OSI?** 2016. Disponível em: <https://www.oficinadanet.com.br/post/15976-o-que-e-o-modelo-osi>. Acesso em: 01 dez. 2020.

SIQUEIRA JUNIOR, A. L. D. **Decomposição de sinais eletromiográficos utilizando Filtros Casados**. 2013. Dissertação (Ciências) — Universidade Federal de Uberlândia, Uberlândia, 2013.

SOFTWARE FOUNDATION PYTHON. **pyqtgraph 0.11.1**. 2020. Disponível em: <https://pypi.org/project/pyqtgraph/>. Acesso em: 30 dez. 2020.

SOFTWARE FOUNDATION PYTHON. **Leitura e escrita de arquivos CSV**. 2021. Disponível em: <https://docs.python.org/pt-br/3/library/csv.html>. Acesso em: 10 jan. 2021.

SOUZA, P. V. E. Sistema de aquisição de sinais de emg e ecg para a plataforma android. Recife, 2015.

STALBER, E.; DIJK, H. V.; FALCK, B.; KIMURA, J.; NEUWIRTH, C.; PITT, M.; PODNAR, S.; RUBIN, D. I.; RUTKOVE, S.; SANDERS, D. B.; SONOO, M.; TANKISI, H.; ZWARTS, M. Standards for quantification of emg and neurography. **Clinical Biomechanics**, n. 130, 2019.

TEXAS INSTRUMENTS. **INA128 - EP Precision, Low Power Instrumentation Amplifiers.** [S.l.], 2005. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/545747/TI1/INA128.html>. Acesso em: 29 out. 2020.

TEXAS INSTRUMENTS. **INA129-EP Precision, Low Power Instrumentation Amplifiers.** [S.l.], 2015. Acesso em: 29 out. 2019.

VELLOSO, R. P. **Protótipo de um eletromiógrafo digital.** 2004. Monografia (Bacharel em Ciências da Computação), Universidade Regional de Blumenau, Blumenau, Brasil.

WEBSTER, J. G. **Medical Instrumentation: Application and design.** 3. ed. New York: John Wiley, 1998.

APÊNDICE A – FIRMWARE DO ESP12F

```
1 //Firmware-Esp-V2.ino
2
3 //Bibliotecas
4 #include <ESP8266WiFi.h>
5 #include <WiFiUdp.h>
6
7 //Configuracoes de conexao WIFI
8 #ifndef STASSID
9 #define STASSID "RedeTeste" //Nome da rede
10 #define STAPSK "123456789" //senha da rede
11 #endif
12
13 //Identificação do Sensor
14 String ident = "S01"; //Identificação do Sensor
15
16 //Pinos Amplificador Digital
17 int PinUD = 4;
18 int PinINC = 2;
19 int cs = 5;
20 int Amp = 4; //Nível de Amplificação (0 a 100);
21 boolean direcao = true;
22
23 //Pinos LED's
24 int LedVerde = 14;
25 int LedVerm = 16;
26 int LedBranco = 12;
27
28 unsigned int localPort = 8080; //Porta de Comunicação
29
30 //Variáveis de leitura
31 int emg = 0;
32 long emg2 = 0;
```

```

33 char ConvertLeitura[4] = {};
34 char vetor[100] = {};
35 String pacote = "";
36 int ReadEMG = 0;
37 int i = 0;
38 int j = 0;
39 int k = 0;
40 int x = 0;
41
42 //Buffer de envio para server UDP
43 char packetBuffer[UDP_TX_PACKET_MAX_SIZE + 1]; //buffer to hold incoming packets
44 char ReplyBuffer[] = "0"; // a string to send back
45
46 WiFiUDP Udp;
47 IPAddress IP_Cliente (192, 168, 137, 1); //IP de acesso
48
49 void setup() {
50   Serial.begin(115200); //Velocidade de Comunicação Serial
51
52   //Definição de Pinos
53   pinMode (cs, OUTPUT);
54   pinMode (PinUD, OUTPUT);
55   pinMode (PinINC, OUTPUT);
56
57   pinMode (LedVerde, OUTPUT);
58   pinMode (LedVerm, OUTPUT);
59   pinMode (LedBranco, OUTPUT);
60
61   LigaLed();
62   ResetAmpDig();
63   DesligaLed();
64
65   //Inicia WIFI
66   WiFi.mode (WIFI_STA);

```

```

67   WiFi.setPhyMode(WIFI_PHY_MODE_11N); // parametro WIFI de velocidade -
68   N = 65Mbps, G = 54Mbps e B=11Mbps
69   WiFi.begin(STASSID, STAPSK);
70   while (WiFi.status() != WL_CONNECTED) {
71       Serial.print('.');
72       delay(500);
73       digitalWrite(LedVerm, HIGH);
74   }
75   Serial.print("Connected! IP address: ");
76   Serial.println(WiFi.localIP());
77   Serial.printf("UDP server on port %d\n", localPort);
78   Udp.begin(localPort);
79 }
80 void loop() {
81     //Led Vermelho indica conexao WIFI
82     if (WiFi.status() == WL_CONNECTED) {
83         digitalWrite(LedVerm, LOW);
84         digitalWrite(LedVerde, HIGH);
85         //Lê Sensor e monta vetor
86
87         pacote = ident;
88         for (i = 0; i < 10; i++) {
89             emg = analogRead(A0);
90             //sprintf(ConverteLeitura, "%d", emg);
91             pacote += String(',');
92             pacote += String(emg);
93             delay(1);
94         }
95
96         pacote.toCharArray(vetor, 100);
97         Serial.println(pacote);
98         //Envia pacote
99         digitalWrite(LedBranco, HIGH);
100        Udp.beginPacket(IP_Cliente, 8080);

```

```

101     //Serial.println(IP_Cliente);
102     Udp.write(vetor);
103     Udp.endPacket();
104     digitalWrite(LedBranco, LOW);
105     //Serial.println("enviado");
106 } else {
107     digitalWrite(LedVerm, HIGH);
108     digitalWrite(LedVerde, LOW);
109     //Serial.println("erro wifi");
110 }
111 //Serial.println(WiFi.status());
112 //Serial.println(WL_CONNECTED);
113
114 }
115
116
117 // AJUSTA A AMPLIFICAAO DO X9C104
118 void ResetAmpDig() {
119     for (j = 0; j < 100; j++) {
120         digitalWrite(PinUD, !direcao);
121         digitalWrite(PinINC, HIGH);
122         delay(20);
123         digitalWrite(cs, LOW);
124         digitalWrite(PinINC, LOW);
125         delay(20);
126         digitalWrite(cs, HIGH);
127         //Serial.println(1);
128     }
129     for (j = (100 - Amp); j > 0; j--) {
130         digitalWrite(PinUD, direcao);
131         digitalWrite(PinINC, HIGH);
132         delay(20);
133         digitalWrite(cs, LOW);
134         digitalWrite(PinINC, LOW);

```

```
135     delay(20);
136     digitalWrite(cs, HIGH);
137     //Serial.println(0);
138 }
139 }
140
141 void DesligaLed() {
142     digitalWrite(LedVerde, LOW);
143     digitalWrite(LedVerm, LOW);
144     digitalWrite(LedBranco, LOW);
145 }
146
147 void LigaLed() {
148     digitalWrite(LedVerde, HIGH);
149     digitalWrite(LedVerm, HIGH);
150     digitalWrite(LedBranco, HIGH);
151 }
```

APÊNDICE B – CÓDIGO DA INTERFACE GRÁFICA

```

1  #Read.me
2
3  # Eletromiograma
4
5  A eletromiografia consiste num exame que avalia a função muscular e
6  diagnostica problemas nervosos ou musculares, a partir dos sinais
7  elétricos liberados pelos mesmos, recolhendo informações acerca da
8  atividade muscular por meio de eletrodos.Nessa aplicação, os dados serão
9  obtidos via conexão UDP definida de por interface de usuário, o qual deve
10  depois de conectar-se, acionar a leitura.
11  Ao receber uma mensagem válida (valor numérico inteiro), o interpretador
12  definirá a mensagem como valor válido para o eletromiograma e o gráfico
13  será atualizado.
14
15  * Leitura de valores enviados pelo sensor (servidor UDP)
16  * Plotagem gráfica dos dados recebidos em "tempo real"
17  * Controle de dados por arquivos de extensão .csv ( $ambiente )
18  * Interpretação de comandos de entrada ( #comando )
19
20
21  ## Desenvolvimento
22  ### Instalação de dependências
23
24  * Python Package Index (update)
25  > python -m pip install --upgrade pip
26
27  * Gráficos e interface científica: PyQtGraph
28  ***http://www.pyqtgraph.org***
29  > pip install pyqtgraph
30
31  * Computação de arrays numéricos multi-dimensionais: NumPy
32  ***https://numpy.org***

```

```
33 > pip install numpy
34
35 * Interface e outras coisas legais: PyQt5
36 ***https://pypi.org/project/PyQt5/**
37 > pip install pyqt5
```

```
1 #Clien.py
2
3 import socket
4 from server import Server
5 import csv
6 import re
7
8 LIM_BUFFER_CSV = 1
9
10 class Client:
11
12     def __init__(self):
13         self.header = "default"
14         self.readyData = []
15         self.readCounter = 0
16
17     def getHeader(self):
18         return self.header
19
20     def setHeader(self, headTitle):
21         self.header = headTitle
22
23     def setReadyData(self, data):
24         self.readyData.append(data)
25
26     def getReadyData(self):
27         return self.readyData
28
```



```
29     def criarConexao(self, ip, port):
30
31         self.server1 = Server()
32
33         if(self.server1.connect(ip, port)):
34
35             print("Connected and listening to", ip, ":", port)
36
37             return True
38
39         else:
40             print("Connection failed")
41             return False
42
43     def desconectar(self):
44         if(self.server1.close()):
45
46             self.server1 = None
47             return True
48
49         else:
50             return False
51
52     def reader(self):
53         try:
54             self.server1.receiveMsg()
55
56             if(self.server1.getData()):
57                 data = self.server1.getData()
58                 # print("msg:", data, type(data))
59
60                 toHandler = self.dataControl(data)
61
62                 return toHandler
```

```

63
64     except:
65         print ("message problem.")
66
67     # Retorna ao plotable o valor ou false
68     def dataControl(self, msg):
69         try:
70             msg2 = msg.split(",")
71
72             self.readCounter = self.readCounter+1
73
74             if(self.readCounter > LIM_BUFFER_CSV):
75                 self.writecsv()
76                 self.clearBuffer()
77
78             return msg2
79
80     except:
81         # Se o valor não for plotável
82         # print("except: ", msg)
83         self.msgHandler(msg)
84         return False
85
86     def msgHandler(self, msg):
87         rTag = r"^#\s" #encontra expressão regular
88         tag = "#\s"
89         rWriteRef = r"^\s" #encontra expressão regular
90         writeRef = "\s"
91
92         # Comando (#)
93         if(bool(re.match(rTag, msg))):
94             msg = msg.replace(tag, "")
95
96         validCmd = self.cmdHandler(msg)

```

```

97
98         # if(not(validCmd)):
99         # print("not a command.")
100
101         # Referência de ambiente csv ($)
102         elif(bool(re.match(rWriteRef, msg))):
103             # msg = msg.replace(writeRef, "")
104             #print("Ambiente: ", msg)
105             self.setHeader(msg)
106
107         # Interpretador de comandos
108         def cmdHandler(self, cmd):
109
110             # cmdList = {
111             #     "test": self.testFn(),
112             #     "start": self.startFn(),
113             #     "save": self.writecsv(),
114             # }
115
116             if(cmd == "con"):
117                 self.server1.sendMessage("connected")
118
119                 return True
120
121             elif(cmd == "save"):
122                 info = self.getHeader() + ".csv"
123
124                 if(self.writecsv()):
125                     info += " salvo."
126                     self.server1.sendMessage(info)
127
128                     return True
129
130             else:

```

```

131         info += " NAO salvo."
132         self.server1.sendMessage(info)
133
134     else:
135         return False
136
137     def writecsv(self):
138         try:
139             fileName = self.getHeader() + ".csv"
140
141             # with open("data.csv", "a", newline='') as file:
142             with open(fileName, "a", newline='') as file:
143                 dataGraph = self.getReadyData()
144                 # writer = csv.writer(file, delimiter=",")
145
146                 writer = csv.writer(file)
147
148                 for i in dataGraph:
149                     # writer.writerow(str(i))
150                     writer.writerow([i])
151
152                 return True
153
154         except:
155             #print("ex writecsv")
156             return False
157
158     def clearBuffer(self):
159         self.readCounter = 0
160         self.readyData = []

```

```
1 #gui.py
```

```
2
```

```
3 from PyQt5 import QtCore, QtGui, QtWidgets
```

```
4 from pyqtgraph import PlotWidget
5 from pyqtgraphptime import time
6 from client import Client
7 import numpy as np
8
9
10 class GUI:
11
12     def __init__(self):
13
14         # self.server1 = None
15         self.client1 = Client()
16
17         # GUI config
18         self.app = QtGui.QApplication([])
19         self.win = QtGui.QMainWindow()
20         self.win.setObjectName("Analise graficos")
21         self.win.resize(1000, 330)
22         self.centralwidget = QtWidgets.QWidget(self.win)
23         self.centralwidget.setObjectName("centralwidget")
24         self.win.setCentralWidget(self.centralwidget)
25         self.timer = QtCore.QTimer()
26         # self.timer.timeout.connect(self.acquireDataGraph1)
27
28         self.timer.timeout.connect(self.graphDataHandler)
29
30         # layout auxs
31         # gridLayout 1
32         self.gridLayout1 = QtWidgets.QWidget(self.centralwidget)
33         self.gridLayout1.setGeometry(QtCore.QRect(10, 140, 201, 173))
34
35         # gridLayout 2
36         self.gridLayout2 = QtWidgets.QWidget(self.centralwidget)
37         self.gridLayout2.setGeometry(QtCore.QRect(10, 330, 201, 173))
```

```
38
39     # labels
40     # graph 1
41     self.labelStatusGraph1 = QtWidgets.QLabel(self.gridLayout1)
42     self.labelStatusGraph1.setObjectName("labelStatusGraph1")
43     self.labelStatusGraph1.setText("Desconectado")
44     self.label2Graph1 = QtWidgets.QLabel(self.gridLayout1)
45     self.label2Graph1.setObjectName("label2Graph1")
46     self.label2Graph1.setText("Status")
47     self.labelPortaGraph1 = QtWidgets.QLabel(self.gridLayout1)
48     self.labelPortaGraph1.setObjectName("labelPortaGraph1")
49     self.labelPortaGraph1.setText("Porta:")
50     self.labelGraph1 = QtWidgets.QLabel(self.gridLayout1)
51     self.labelGraph1.setObjectName("labelGraph1")
52     self.labelGraph1.setText("Ambiente: $" + self.client1.getHeader())
53     self.labelIpGraph1 = QtWidgets.QLabel(self.gridLayout1)
54     self.labelIpGraph1.setObjectName("labelIpGraph1")
55     self.labelIpGraph1.setText("IP:")
56
57     # Banner
58     self.labelImage = QtWidgets.QLabel(self.centralwidget)
59     self.labelImage.setGeometry(QtCore.QRect(0, 0, 1041, 121))
60     self.labelImage.setContextMenuPolicy(QtCore.Qt.PreventContextMenu)
61     self.labelImage.setFrameShape(QtWidgets.QFrame.StyledPanel)
62     self.labelImage.setTextFormat(QtCore.Qt.AutoText)
63     self.labelImage.setPixmap(QtGui.QPixmap("./BANNER.PNG"))
64     self.labelImage.setObjectName("labelImage")
65
66     # buttons
67     # graph 1
68     self.btnIniciarGraph1 = QtWidgets.QPushButton(self.gridLayout1)
69     self.btnIniciarGraph1.setObjectName("btnIniciarGraph1")
70     self.btnIniciarGraph1.setText("Iniciar")
71     self.btnIniciarGraph1.setEnabled(False)
```

```
72     self.btnConectarGraph1 = QtWidgets.QPushButton(self.gridLayout1)
73     self.btnConectarGraph1.setObjectName("btnConectarGraph1")
74     self.btnConectarGraph1.setText("Conectar")
75     self.btnSalvarGraph1 = QtWidgets.QPushButton(self.gridLayout1)
76     self.btnSalvarGraph1.setObjectName("btnSalvarGraph1")
77     self.btnSalvarGraph1.setText("Salvar")
78     self.btnSalvarGraph1.setEnabled(False)
79     self.btnDesconectarGraph1 = QtWidgets.QPushButton(self.gridLayout1)
80     self.btnDesconectarGraph1.setObjectName("btnDesconectarGraph1")
81     self.btnDesconectarGraph1.setText("Desconectar")
82     self.btnDesconectarGraph1.setEnabled(False)
83     self.btnPararGraph1 = QtWidgets.QPushButton(self.gridLayout1)
84     self.btnPararGraph1.setObjectName("btnPararGraph1")
85     self.btnPararGraph1.setText("Parar")
86     self.btnPararGraph1.setEnabled(False)
87     self.btnEncerrarGraph1 = QtWidgets.QPushButton(self.gridLayout1)
88     self.btnEncerrarGraph1.setObjectName("btnEncerrarGraph1")
89     self.btnEncerrarGraph1.setText("Encerrar")
90     self.btnEncerrarGraph1.setEnabled(False)
91
92     # inputs
93     # graph 1
94     self.inputIpGraph1 = QtWidgets.QLineEdit(self.gridLayout1)
95     self.inputIpGraph1.setObjectName("inputIpGraph1")
96     self.inputPortaGraph1 = QtWidgets.QLineEdit(self.gridLayout1)
97     self.inputPortaGraph1.setObjectName("inputPortaGraph1")
98     self.inputIpGraph1.setObjectName("inputIpGraph1")
99
100    # controlBox1
101    self.ControlGraphBox1 = QtWidgets.QGridLayout(self.gridLayout1)
102    self.ControlGraphBox1.setContentsMargins(0, 0, 0, 0)
103    self.ControlGraphBox1.setObjectName("ControlGraphBox1")
104
105    self.ControlGraphBox1.addWidget(
```

```

106         self.labelStatusGraph1, 1, 1, 0, 0, QtCore.Qt.AlignHCenter)
107
108     self.ControlGraphBox1.addWidget(
109         self.label2Graph1, 1, 0, 1, 1, QtCore.Qt.AlignHCenter)
110     self.ControlGraphBox1.addWidget(
111         self.labelGraph1, 0, 0, 1, 1, QtCore.Qt.AlignRight)
112     self.ControlGraphBox1.addWidget(
113         self.labelIpGraph1, 2, 0, 1, 1, QtCore.Qt.AlignHCenter)
114     self.ControlGraphBox1.addWidget(self.inputIpGraph1, 2, 1, 1, 1)
115     self.ControlGraphBox1.addWidget(
116         self.labelPortaGraph1, 3, 0, 1, 1, QtCore.Qt.AlignHCenter)
117     self.ControlGraphBox1.addWidget(self.inputPortaGraph1, 3, 1, 1, 1)
118
119     self.ControlGraphBox1.setColumnStretch(1, 1+1)
120
121     self.ControlGraphBox1.addWidget(self.btnConectarGraph1, 4, 0, 1, 1)
122     self.ControlGraphBox1.addWidget(self.btnDesconectarGraph1, 4, 1, 1, 1)
123     self.ControlGraphBox1.addWidget(self.btnIniciarGraph1, 5, 0, 1, 1)
124     self.ControlGraphBox1.addWidget(self.btnPararGraph1, 5, 1, 1, 1)
125     self.ControlGraphBox1.addWidget(self.btnEncerrarGraph1, 6, 1, 1, 1)
126     self.ControlGraphBox1.addWidget(self.btnSalvarGraph1, 6, 0, 1, 1)
127
128     # graphView1
129     self.graphicsView1 = PlotWidget(self.centralwidget)
130     self.graphicsView1.setGeometry(QtCore.QRect(230, 140, 759, 171))
131     self.graphicsView1.setObjectName("graphicsView1")
132     self.graphicsView1.setRange(yRange=[-20,20], padding=None,
133     update=False) #Ajusta escala Y
134     #self.graphic1 = self.graphicsView1.plot(self.dataTempGraph1)
135     self.graphicsView1.setLabel('bottom','Tempo', units=" Segundos" )
136     self.graphicsView1.setLabel('left','Amplitude', units="mV" )
137     # linhas de separacao
138     # linha 1
139     self.line1 = QtWidgets.QFrame(self.centralwidget)

```



```
140     self.line1.setGeometry(QRect(10, 120, 981, 20))
141     self.line1.setFrameShape(QtWidgets.QFrame.HLine)
142     self.line1.setFrameShadow(QtWidgets.QFrame.Sunken)
143     self.line1.setObjectName("line1")
144
145     # data Graph
146     self.dataGraph1 = [0]
147
148     self.dataTempGraph1 = np.zeros(5000, dtype=int)
149     # position
150     self.position1 = 0
151
152     self.graphic1 = self.graphicsView1.plot(self.dataTempGraph1)
153
154     # buttons actions
155     # Conectar
156     self.btnConectarGraph1.clicked.connect(lambda: self.tryConnect())
157
158     # Desconectar
159     self.btnDesconectarGraph1.clicked.connect(
160         lambda: self.tryDisconnect())
161
162     # Iniciar
163     self.btnIniciarGraph1.clicked.connect(lambda: self.iniciarGraph1())
164
165     # parar
166     self.btnPararGraph1.clicked.connect(lambda: self.paraGraph1())
167
168     self.btnSalvarGraph1.clicked.connect(lambda: self.trySave())
169
170     # Encerrar
171     self.btnEncerrarGraph1.clicked.connect(lambda: self.encerrarGraph1())
172
173     #Fps
```

```
174         self.timepac = None
175         self.lastTime = time()
176
177     def tryConnect(self):
178         input_ip = str(self.inputIpGraph1.text())
179         input_port = int(self.inputPortaGraph1.text())
180
181         # Ip local (Conexão rápida - dev):
182         #input_ip = str("192.168.137.1")
183         #input_port = int("8080")
184         connected = self.client1.criarConexao(input_ip, input_port)
185
186         if(connected):
187             self.labelStatusGraph1.setText("Pronto")
188
189             self.btnConectarGraph1.setEnabled(False)
190             self.btnIniciarGraph1.setEnabled(True)
191             self.btnDesconectarGraph1.setEnabled(True)
192             self.inputIpGraph1.setReadOnly(True)
193             self.inputPortaGraph1.setReadOnly(True)
194         else:
195             print("Nao conectado")
196
197     def tryDisconnect(self):
198
199         disconnected = self.client1.desconectar()
200
201         if(disconnected):
202             self.labelStatusGraph1.setText("Desconectado")
203             self.btnConectarGraph1.setEnabled(True)
204             self.btnDesconectarGraph1.setEnabled(False)
205             self.btnIniciarGraph1.setEnabled(False)
206             self.inputIpGraph1.setReadOnly(False)
207             self.inputIpGraph1.setText("")
```

```
208         self.inputPortaGraph1.setReadOnly(False)
209         self.inputPortaGraph1.setText("")
210
211     def iniciarGraph1(self):
212         self.labelStatusGraph1.setText("Lendo dados...")
213         self.btnDesconectarGraph1.setEnabled(False)
214         self.btnIniciarGraph1.setEnabled(False)
215         self.btnSalvarGraph1.setEnabled(False)
216         self.btnEncerrarGraph1.setEnabled(False)
217         self.btnPararGraph1.setEnabled(True)
218         self.startPlotting()
219
220     def paraGraph1(self):
221         self.labelStatusGraph1.setText("Leitura parada...")
222         self.btnPararGraph1.setEnabled(False)
223         self.btnIniciarGraph1.setEnabled(True)
224         self.btnSalvarGraph1.setEnabled(True)
225         self.btnEncerrarGraph1.setEnabled(True)
226         self.app.processEvents()
227         self.stopPlotting()
228
229     def encerrarGraph1(self):
230         self.tryDisconnect()
231         self.btnSalvarGraph1.setEnabled(False)
232         self.btnEncerrarGraph1.setEnabled(False)
233         self.dataGraph1 = [0]
234         self.dataTempGraph1 = np.zeros(10)
235         self.position1 = 0
236         self.graphic1.setData(self.dataTempGraph1)
237         self.graphic1.setPos(self.position1, 0)
238
239     def updateGraph1(self):
240         self.graphic1.setData(self.dataTempGraph1)
241         self.graphic1.setPos(self.position1, 0)
```

```

242     self.position1 += 10 #AJUSTA PASSO DA ESCALA
243     self.dataGraph1.append(self.dataTempGraph1[-1])
244     self.graphic1.updateItems()
245     self.graphicsView1.updateMatrix()
246
247     def startPlotting(self):
248         self.timer.start(1)
249
250     def stopPlotting(self):
251         self.timer.stop()
252
253     def acquireDataGraph1(self, data):
254         # print("adq: ", data)
255
256         for valor in data:
257             try:
258                 calc = (int(valor)) - 468 #/Remover OffSet
259                 plotable = (((calc*3.3)/1024)/204)*10000/2
260                 self.dataTempGraph1[:-1] = self.dataTempGraph1[1:]
261                 self.dataTempGraph1[-1] = plotable
262
263                 self.client1.setReadyData(plotable)
264
265             except:
266                 #print(valor)
267                 self.client1.msgHandler(valor)
268
269         self.updateGraph1()
270         self.TX_PACOTE_SHOW()
271
272     def graphDataHandler(self):
273         vetor = self.client1.reader()
274
275         if(vetor):

```

```

276         self.acquireDataGraph1(vetor)
277
278     def trySave(self):
279
280         #print(self.client1.getReadyData())
281
282         fileName = self.client1.getHeader() + ".csv"
283
284         self.labelStatusGraph1.setText("Salvando " + fileName)
285
286         if(self.client1.writecsv()):
287             self.labelStatusGraph1.setText(fileName + " salvo.")
288
289         else:
290             self.labelStatusGraph1.setText(fileName + " não salvo.")
291
292     def start(self):
293         self.win.show()
294         self.app.exec_()
295
296     def TX_PACOTE_SHOW(self):
297         now = time()
298         dt = now - self.lastTime
299         self.lastTime = now
300         if self.timepac is None:
301             self.timepac = 1.0/dt
302         else:
303             s = np.clip(dt*3., 0, 1)
304             self.timepac = self.timepac * (1-s) + (1.0/dt) * s
305         self.graphicsView1.setTitle('%0.2f fps' % self.timepac)
306         print ("%d",self.timepac)

```

```

1 #server.py

```

```

2

```

```
3 import socket
4
5
6 class Server:
7     def __init__(self):
8         self.data = None
9         self.sock = None
10        self.onlineIp = None
11        self.onlinePort = None
12        self.target = None
13
14    def getData(self):
15        return self.data
16
17    def setData(self, msg):
18        self.data = msg
19
20    def getOnlineIp(self):
21        return self.onlineIp
22
23    def setOnlineIp(self, ip):
24        self.onlineIp = ip
25
26    def getOnlinePort(self):
27        return self.onlinePort
28
29    def setOnlinePort(self, port):
30        self.onlinePort = port
31
32    def setTarget(self, addr):
33        self.target = addr
34
35    def getTarget(self):
36        return self.target
```

```
37
38     def close(self):
39         try:
40             self.sock.close()
41             return True
42         except:
43             return False
44
45     def connect(self, ip, porta):
46         try:
47             # SOCK_DGRAM == UDP CONNECTION
48             self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, 0)
49             self.sock.setblocking(False)
50             self.sock.bind((ip, porta))
51
52             self.setOnlineIp = ip
53             self.setOnlinePort = porta
54
55             return True
56
57         except:
58             self.setOnlineIp = None
59             self.setOnlinePort = None
60
61             return False
62
63     def receiveMsg(self):
64         try:
65             msg, addr = self.sock.recvfrom(128)
66             msg = msg.decode("utf-8")
67
68             self.setTarget(addr)
69
70             if(msg != None):
```

```
71         #print("accomplished:", msg)
72         self.setData(msg)
73         return True
74
75     except:
76         self.setData(False)
77
78     def sendMessage(self, msg):
79         try:
80             # print("sending \"", msg, "\"to:", self.getTarget())
81             ret = self.sock.sendto(msg.encode('utf-8'), self.getTarget())
82
83         except:
84             print("not sent.")
```

```
1  #main.py
2
3  from gui import GUI
4
5
6  if(__name__ == "__main__"):
7      window = GUI()
8      window.start()
```
