

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA
PPGSE – PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE ENERGIA

LILIAN VASELQUIU MULLER

**SISTEMA DE MONITORAMENTO DA FREQUÊNCIA
CARDÍACA VIA LÓGICA FUZZY EM BICICLETA
ERGOMÉTRICA COM MICROGERAÇÃO DE ENERGIA**

DISSERTAÇÃO DE MESTRADO

CURITIBA
2019

LILIAN VASELQUIU MULLER

**SISTEMA DE MONITORAMENTO DA FREQUÊNCIA
CARDÍACA VIA LÓGICA FUZZY EM BICICLETA
ERGOMÉTRICA COM MICROGERAÇÃO DE ENERGIA**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Sistemas de Energia da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Mestre em Engenharia Elétrica” – Área de Concentração: Automação e Sistemas de Energia.

Orientador: Prof. Dr. Paulo Cícero Fritzen

CURITIBA

2019

Dados Internacionais de Catalogação na Publicação

Müller, Lilian Vaselqui

Sistema de monitoramento da frequência cardíaca via lógica fuzzy em bicicleta ergométrica com microgeração de energia [recurso eletrônico] / Lilian Vaselqui Müller.-- 2019.

1 arquivo eletrônico (114 f.) : PDF ; 2,18 MB.

Modo de acesso: World Wide Web.

Texto em português com resumo em inglês.

Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Sistemas de Energia. Linha de Pesquisa: Automação e Sistemas de Energia, Curitiba, 2019.

Bibliografia: f. 91-94.

1. Sistemas de energia elétrica - Dissertações. 2. Batimento cardíaco - Monitorização. 3. Lógica difusa. 4. Exercícios físicos - Equipamento e acessórios. 5. Bicicletas. 6. Energia - Fontes alternativas. 7. Sistemas especialistas (Computação) - Desenvolvimento. 8. Arduino (Controlador programável). 9. MATLAB (Programa de computador). 10. Métodos de simulação. I. Fritzen, Paulo Cícero, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Sistemas de Energia. III. Título.

CDD: Ed. 23 -- 621.31

TERMO DE APROVAÇÃO DE DISSERTAÇÃO

A Dissertação de Mestrado intitulada “Sistema de monitoramento da frequência cardíaca via Lógica Fuzzy em bicicleta ergométrica com microgeração de energia”, defendida em sessão pública pelo(a) candidato(a) Lilian Vaselqui Muller, no dia 22 de maio de 2019, foi julgada para a obtenção do título de Mestre em Engenharia Elétrica, área de concentração Automação e Sistemas de Energia, e aprovada em sua forma final, pelo Programa de Pós-Graduação em Sistemas de Energia.

BANCA EXAMINADORA:

Prof. Dr. Paulo Cícero Fritzen - Presidente – UTFPR
Profa. Dra. Mariana Antônia Aguiar Furucho – UTFPR
Prof. Dr. Wilerson Sturm – IFPR

A via original deste documento encontra-se arquivada na Secretaria do Programa, contendo a assinatura da Coordenação após a entrega da versão corrigida do trabalho.

Curitiba, 22 de maio de 2019.

Carimbo e Assinatura do(a) Coordenador(a) do Programa

AGRADECIMENTOS

Expresso aqui o meu mais profundo agradecimento a todos, que diretamente ou indiretamente, tornaram possível o desenvolvimento deste trabalho.

Primeiramente agradeço a Deus, e seu filho enviado a nós, Jesus Cristo, pela força dada a cada dia para enfrentar todas as dificuldades encontradas.

Agradeço ao meu orientador professor Doutor Paulo Cícero Fritzen, pelo apoio, paciência e orientação relativa ao desenvolvimento deste trabalho.

A grande amiga Marjorie Belinelli, minha Mãe Terezinha e ao meu pai Renato (in memoriam), por me apoiarem em todos os momentos durante esta jornada.

Agradeço aos meus amigos Carlos Sakiti e Leonardo Weiss, pelo apoio, acesso e disposição de informações técnicas.

Um agradecimento especial ao Prof. Msc Ednilson Soares Maciel e à Profa. Dra. Mariana Antonia Aguiar Furucho pela ajuda técnica no desenvolvimento do sistema.

À banca examinadora pelo tempo dedicado para a leitura deste trabalho e pelas sugestões.

Por último, um grande obrigado a todos os professores PPGSE -Programa de Pós-graduação em Sistemas de Energia, a todos os demais amigos, parentes e colegas de trabalho que ao longo dos meses de realização deste trabalho, demonstraram todo o apoio e incentivo, que me permitiu a conclusão desta dissertação.

“Que ninguém se engane, só se consegue a simplicidade através de muito trabalho”

Clarice Lispector, 1977

RESUMO

MÜLLER, Lilian Vasselquiu. Sistema de Monitoramento da Frequência Cardíaca Via Lógica Fuzzy em Bicicleta Ergométrica com Microgeração de Energia. 114 páginas. Dissertação para obtenção do Título de Mestre em Engenharia Elétrica – Programa de Pós-Graduação em Sistemas de Energia, Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

Esta pesquisa apresenta o desenvolvimento de um sistema de recomendação baseado em lógica *Fuzzy*, o qual monitora os batimentos cardíacos do usuário e indica a rotação adequada conforme a frequência cardíaca e a carga mensuradas. Em conjunto com o sistema de recomendação foi feito um sistema de microgeração de energia renovável por meio de aplicação de esforços físicos em uma bicicleta ergométrica. Conectou-se na bicicleta ergométrica, instalada no laboratório de máquinas elétricas do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná-UTFPR Curitiba, um alternador CA ao volante de inércia da bicicleta, gerando assim energia elétrica de acordo com o esforço aplicado ao pedal. O objetivo do sistema desenvolvido é de assegurar que o ciclista gere energia elétrica sustentável, por um determinado período, com segurança a sua saúde. Para a construção do sistema de monitoramento de batimento cardíaco foi utilizado o ARDUINO® que é uma plataforma de prototipagem eletrônica que utiliza um microcontrolador AT Mega 2560. A placa desta plataforma se chama Arduino Mega 2560. O dispositivo criado, *hardware* e *software*, são apresentados de forma detalhada e os resultados dos testes realizados com 3 usuários (praticantes entre 23 e 40 anos) mostraram a rotação adequada para cada um e que também as frequências cardíacas destes três praticantes se mantiveram entre mínima, ideal e máxima, garantindo deste modo, a integridade física destes usuários na geração de energia sustentável. Os valores atingidos nos testes pelos praticantes foram comparados a valores simulados no software MATLAB®.

Palavras-Chave: Energia Renovável. Lógica *Fuzzy*. Microgeração de Energia Elétrica. Rotação. Sistema de Recomendação.

ABSTRACT

MÜLLER, Lilian Vasselquiu. Heart Rate Monitor System by means of Fuzzy Logic in Ergometric Bicycle with Energy Microgeneration. 114 pages. Dissertation for obtaining the Master's Degree in Electrical Engineering - Post-Graduation Program in Energy Systems, Federal Technological University of Paraná. Curitiba, 2019.

This research presents the development of a Fuzzy logic-based recommendation system that supervises the user's heart rate to indicate the adequate rotation according to the measured heart rate and load. Simultaneously with the recommendation system, a micro-generation renewable energy system was made through physical exertion on a stationary bicycle. It was coupled to an ergometric bicycle, which was installed in the laboratory of electrical machines of the DAELT (Department of Electrotechnology) of the Federal Technological University of Paraná -UTFPR Curitiba, an alternator AC to on the inertia flywheel, thus generating electric energy according to the effort applied to the pedal. In order to ensure that the cyclist manages sustainable electric energy, for a certain period and with safety to his health. For the construction of the heart rate monitoring system, ARDUINO® was used as an electronic prototyping platform that uses an AT Mega 2560 microcontroller. The board of this platform is called Arduino Mega 2560. The device created, hardware and software, are presented in detail and the results of the tests performed with 3 users (practitioners between 23 and 40 years) showed the adequate rotation for each one and that the heart rates of these three practitioners remained between minimum, optimal and maximum, thus ensuring the physical integrity of these users in sustainable power generation. The values achieved in the tests by the practitioners were compared to simulated values in the MATLAB® software.

Keywords: Fuzzy Logic. Micro Generation of Electrical Energy. Recommendation System. Renewable Energy. Rotation.

LISTA DE FIGURAS

Figura 1- Aspectos construtivos do gerador síncrono.....	23
Figura 2 - Sistema Lógico Fuzzy.....	26
Figura 3 - Função de Pertinência para um conjunto Nebuloso.....	28
Figura 4- Configuração básica de um Sistema Fuzzy.....	36
Figura 5 - Protocolo de Pesquisa.....	42
Figura 6 - Bicicleta Ergométrica com Sistema de recomendação e de microgeração de energia.....	45
Figura 7- Sistema de Transmissão de Movimento.....	46
Figura 8 - Esquemático de funcionamento do sistema.....	49
Figura 9 - Placa de Integração.....	50
Figura 10 - Diagrama da placa de Integração.....	51
Figura 11 - ARDUÍNO® Mega 2560.....	52
Figura 12 – Tela 1 display para ARDUÍNO® 16X2.....	54
Figura 13 – Tela 2 display para ARDUÍNO® 16X2.....	54
Figura 14 – Tela 3 display para ARDUÍNO® 16X2.....	55
Figura 15 - Filtro LC de modo comum.....	55
Figura 16- Sensor efeito Hall.....	56
Figura 17- Sensor de medição de frequência cardíaca	57
Figura 18 - Drive de proteção dos pinos de saída do ARDUÍNO®	58
Figura 19 - Configuração básica do uso da ferramenta Fuzzy Logical Toolbox com as especificações estabelecidas.....	59
Figura 20 - Fluxograma do software de Apoio a Decisão.....	61
Figura 21 - Variável de entrada batimentos cardíacos por minuto (entre 50 e 220 batimentos por minuto).....	64
Figura 22 - Variável de entrada carga acoplada ao gerador universo de discurso entre 0 e 300 watts.....	64
Figura 23 - Variável de saída rotação. Universo de discurso entre 0 a 100 rotações por minuto.....	66
Figura 24 – Base de Regras no Matlab.....	69
Figura 25 - Desfuzzificação utilizando o método do centro de área.....	71
Figura 26 - Pontos para Conexão dos Eletrodos no Corpo Humano.....	75

Figura 27- Função de Pertinência da Frequência Cardíaca do usuário U1.....	78
Figura 28 - Função de Pertinência da carga do usuário U1.....	78
Figura 29 - Função de Pertinência da rotação do usuário U1.....	78
Figura 30 - Superfície gerada para perfil do usuário U1.....	79
Figura 31 - Saída1 do usuário U1.....	79
Figura 32 - Saída 2 do usuário U1.....	80
Figura 33 - Saída 3 do usuário U1.....	80
Figura 34 - Função de Pertinência FC do usuário U2.....	81
Figura 35 - Função de Pertinência da carga dos usuários U2 e U3.....	81
Figura 36 - Função de Pertinência da rotação usuários U2 e U3.....	82
Figura 37 - Superfície gerada para os perfis dos usuários U2 e U3.....	82
Figura 38 - Saída 1 usuário U2.....	83
Figura 39- Saída 2 usuário U2.....	83
Figura 40 -Saída 3 usuário U2.....	84
Figura 41 - Função de Pertinência da frequência cardíaca do usuário U3.....	85
Figura 42 - Saída 1 usuário U3.....	85
Figura 43- Saída 2 usuário U3.....	86
Figura 44 -Saída 3 usuário U3.....	86

LISTA DE QUADROS

Quadro 1 – Função de Pertinência Utilizada na Lógica Fuzzy.....	29
Quadro 2 - Classificação das Frequências Cardíacas: Mínima, Ideal e Máxima.....	38
Quadro 3 – Classificação da metodologia (resumo).....	40
Quadro 4 - Perfil dos Usuários.....	72

LISTA DE TABELAS

Tabela 1 - Valores nominais do gerador síncrono AC.....	46
Tabela 2 - Mapa de Regras Fuzzy.....	68
Tabela 3 - Valores Obtidos nos Testes Usuário 1.....	75
Tabela 4 - Valores Obtidos nos Testes Usuário 2.....	75
Tabela 5 - Valores Obtidos nos Testes Usuário 3.....	75
Tabela 6 - Valores de saída do Sistema de Recomendação Usuário 1.....	76
Tabela 7 - Valores de saída do Sistema de Recomendação Usuário 2.....	76
Tabela 8 - Valores de saída do Sistema de Recomendação Usuário 3.....	76
Tabela 9 - Comparação entre a saída do controlador ARDUÍNO® e o MATLAB® para o usuário U1.....	87
Tabela 10 - Comparação entre a saída do controlador ARDUÍNO® e o MATLAB® para o usuário U2.....	87
Tabela 11 - Comparação entre a saída do controlador ARDUÍNO® e o MATLAB® para o usuário U3.....	87

LISTA DE SIGLAS, ABREVIATURAS E ACRÔNIMOS

ANEEL	Agência Nacional de Energia Elétrica
BPM	Batimento por Minuto
CA	Corrente Alternada
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CC	Corrente Contínua
CT	Curitiba
DAELT	Departamento de Eletrotécnica
FC	Frequência Cardíaca
FEM	Força Eletromotriz
IEA	Agência Internacional de Energia
IEEE XPLORE	Institute of Electrical and Eletronics Engineers explorer
kW	Kilowatts
SOBRAC	Segundo a Sociedade Brasileira de Arritmias Cardíacas
RN	Resolução Normativa
RPM	Rotação por Minuto
U1	Usuário 1
U2	Usuário 2
U3	Usuário 3
UTFPR	Universidade Tecnológica Federal do Paraná
V	Volts
W	Watts

SUMÁRIO

1 INTRODUÇÃO	15
1.1 CONSIDERAÇÕES INICIAIS	15
1.2 PROBLEMA DE PESQUISA	16
1.3 OBJETIVOS	18
1.3.1 Objetivo Geral	18
1.3.2 Objetivos Específicos	18
1.4 JUSTIFICATIVA	18
1.5 ESTRUTURA DO TRABALHO	20
2 FUNDAMENTAÇÃO TEÓRICA	21
2.1 MICROGERAÇÃO DE ENERGIA ELÉTRICA PELA PRÁTICA DE EXERCÍCIOS FÍSICOS	21
2.2 GERADOR SÍNCRONO	23
2.2.1 Gerador síncrono de imã permanente	25
2.3 LÓGICA FUZZY	25
2.3.1 Conjuntos <i>Fuzzy</i>	27
2.3.2 Operações em Conjuntos Nebulosos	29
2.3.2.1 Operadores s-norma	30
2.3.2.2 Operadores t-norma	31
2.3.2.3 Propriedades Algébricas utilizando as definições de união e intersecção	31
2.3.2.4 Variáveis Linguísticas	32
2.3.2.5 Operadores de Implicação e proposições <i>Fuzzy</i>	33
2.3.3 Sistema <i>Fuzzy</i>	34
2.3.3.1 Estrutura de um sistema <i>Fuzzy</i>	35
2.4 MEDIÇÃO DA FREQUÊNCIA CARDÍACA	37
3 MATERIAIS E MÉTODOS	39
3.1 CLASSIFICAÇÃO DA METODOLOGIA	39
3.2 PROTOCOLO DE PESQUISA	40
4 DESENVOLVIMENTO DO TRABALHO	44
4.1 SISTEMA DE APOIO À DECISÃO E DE GERAÇÃO DE ENERGIA SUSTENTÁVEL	44
4.2 ADAPTAÇÃO DO GERADOR (CA)	44

4.3 COMPONENTES DO SISTEMA DE APOIO À DECISÃO BASEADO EM LÓGICA FUZZY	48
4.3.1 Controlador ARDUINO® Mega 2560.....	52
4.3.2 Display Digital e Interface com o usuário	53
4.3.3 Filtro de Modo Comum	55
4.3.4 Sensores: Sensor de Efeito Hall e Sensor de Frequência Cardíaca	56
4.3.5 Drive de Acionamento das Cargas.....	57
4.4 SIMULAÇÃO DO SISTEMA DE APOIO À DECISÃO NO MATLAB	59
4.5 IMPLANTAÇÃO DA LÓGICA FUZZY NA PLATAFORMA ARDUINO	60
4.5.1 Estrutura do Programa de Apoio a Decisão Baseado em Lógica Fuzzy	60
4.5.2 Entradas e Saída do Sistema.....	63
4.5.3 Regras <i>Fuzzy</i>	67
4.6 COLETA DOS DADOS.....	72
5 RESULTADOS E DISCUSSÕES	74
5.1 RESULTADOS DOS TESTES NO LABORATÓRIO.....	74
6 CONCLUSÃO E RECOMENDAÇÕES PARA TRABALHOS FUTUROS.....	89
6.1 CONCLUSÕES	89
6.2 DIFICULDADES ENCONTRADAS E RECOMENDAÇÕES PARA TRABALHOS FUTUROS	90
REFERÊNCIAS.....	91
APÊNDICE A.....	95

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

De acordo com Araújo e Araújo (2000), atividade física é conceituada como todo movimento corporal com gasto energético acima dos níveis de repouso.

Aliado ao conceito de atividade física, tem-se o conceito de qualidade de vida e segurança do praticante de exercício físico durante a atividade. A prática de ciclismo, tanto em academias, quanto ao livre ou correr alguns quilômetros demandam uma habilidade e um gasto energético consideravelmente maior comparado aos demais exercícios em academias e práticas de exercícios em geral (ARAÚJO; ARAÚJO, 2000). Este esforço físico maior na prática do ciclismo ou corrida podem acarretar problemas cardíacos e por este motivo, a realização de check-ups cardíacos constantes antes do início das atividades físicas são essenciais (SOBRAC,2018).

Outro ponto importante é ter a informação do batimento cardíaco no momento da prática do exercício físico, classifica-lo como mínimo, ideal e máximo e também da orientação da carga de exercício adequada, pois a prática de esportes intensos, por um longo período de tempo, estressam o coração e podem causar danos.

Segundo a Sociedade Brasileira de Arritmias Cardíacas (SOBRAC), 20% da população tem algum tipo de arritmia. E um levantamento junto com o Ministério da saúde 80 a 90% das mortes súbitas estão relacionadas as arritmias cardíacas (SOBRAC,2018). Também, conforme a SOBRAC, 2018, o coração é órgão que tem um ritmo regular, compassado, qualquer alteração na formação ou condução desse estímulo pode provocar alterações no ritmo cardíaco caracterizando a arritmia cardíaca (alteração ou distúrbio do ritmo cardíaco normal).

As alterações do ritmo cardíaco ou das conduções dos estímulos podem ser letais (morte súbita), ser sintomáticas (síncope, tonturas, palpitações, etc.) ou assintomáticas, e , com as arritmias, o coração pode não ser capaz de bombear sangue suficiente para o corpo, o que pode danificar cérebro, outros órgãos e o próprio coração (SOBRAC,2018).

Portanto, se faz necessário desenvolver métodos e sistemas para identificar a faixa adequada do batimento cardíaco e determinar a carga de trabalho adequada para prática do ciclismo de forma segura à saúde do praticante.

Neste contexto, da prática de exercícios físicos, a energia gasta pelas pessoas durante os exercícios na academia é perdida em forma de calor, e esta energia biomecânica perdida poderia ser convertida em uma forma de energia útil, economizando recursos energéticos (CHALERMTHAI et.al., 2015).

As academias ou centros esportivos consomem energia elétrica não somente para alimentar alguns equipamentos (como esteiras, bicicletas entre outros), mas também utilizam a eletricidade para satisfazer o conforto e bem-estar dos seus frequentadores, empregando energia elétrica em: aparelhos de som, televisores, sistema de iluminação e climatização adequadas, chuveiros, etc (CHALERMTHAI et.al., 2015).

Para se tornarem sustentáveis, muitas academias já fazem investimentos na tecnologia do sistema de iluminação, a fim de diminuir o consumo de energia. Porém, há escassez em estudos que façam o levantamento do consumo de cada equipamento (empregado na prática de exercícios), e o quanto cada um pode contribuir na geração de energia elétrica por meio da transformação da energia mecânica gerada pelo esforço físico dos praticantes.

A instalação de geradores e baterias nos(as): correias de esteiras, pedais de bicicleta, máquinas elípticas, entre outros equipamentos, pode ser uma das soluções para gerar e armazenar energia elétrica, a qual pode ser empregada nas demais instalações das academias, racionando o consumo de recursos energéticos (CHALERMTHAI; SADA; SARFRAZ; RADI; 2015).

Assim, este trabalho desenvolveu um sistema para classificar o batimento cardíaco do usuário, determinar a rotação adequada para prática do ciclismo, com segurança ao praticante, em função do ritmo cardíaco, juntamente com a geração de energia elétrica, por meio de esforços aplicados em um pedal de uma bicicleta ergométrica (transformação da energia mecânica em elétrica).

1.2 PROBLEMA DE PESQUISA

Apesar da prática de exercícios físicos de forma moderada e regular conferir muitos benefícios cardiovasculares para a saúde, a atividade física intensa e/ou em excesso pode aumentar o risco de anormalidades cardíacas, fator de risco de morbidade ou mortalidade em longo prazo (SEIDL; ASPLUND, 2014). Avarias cardiológicas ligadas à prática de exercícios físicos estão relacionadas a cardiopatia

orgânica e muitos atletas desconhecem serem cardiopatas por não apresentarem sintomas, por isso é difícil prevenir uma morte súbita durante um exercício. (SEIDL; ASPLUND,2014).

O crescimento de risco de morte súbita em adolescentes e jovens adultos aumentam significativamente (2,8 vezes) quando associados a esforços físicos intensos durante competições esportivas (MARON; PELLICCIA, 2006). O exercício atua como um fator desencadeante de taquiarritmias ventriculares letais ocasionadas pela doença cardíaca preexistente e que não foi diagnosticada (MARON; PELLICCIA, 2006). A maior causa de mortes súbitas em jovens atletas treinados de acordo com as pesquisas feitas em autópsias são provocadas por doenças cardiovasculares congênitas e clinicamente silenciosas (MARON; PELLICCIA, 2006).

A atividade física de pedalar exige do ciclista um esforço cardíaco acima do comum, onde são combinados exercícios de resistência e força (MARON; PELLICCIA, 2006). Como resultado da atividade física intensa ocorre o aumento moderado do consumo de oxigênio e aumentos substanciais da pressão arterial, da resistência vascular periférica e da frequência cardíaca (MARON; PELLICCIA, 2006).

Desta forma, para proporcionar mais segurança para o ciclista quando está pedalando na bicicleta ergométrica, por um determinado período, os batimentos cardíacos devem ser monitorados e classificados bem como a intensidade de esforço físico feito pelo praticante.

Assim, o sistema desenvolvido visa monitorar o batimento cardíaco e estabelecer a rotação adequada para prática do ciclismo, orientando o praticante em relação ao limite de batimento e carga máxima que ele pode atingir durante o exercício, prevenindo alguma anomalia cardíaca.

As academias ou centros esportivos consomem energia elétrica para alimentar alguns equipamentos utilizados para exercícios físicos (como esteiras, bicicletas, entre outros), os quais poderiam ser substituídos por equipamentos que reutilizem a energia gastas pelas pessoas (energia mecânica) durante as atividades físicas.

Bicicletas ergométricas podem converter a energia mecânica em elétrica para abastecer as demais instalações das academias. Estudos feitos na universidade de Oxford mostraram que um ciclista poderia facilmente produzir em média 75 watts de potência elétrica sustentável com uma tensão de 12 volts e velocidade de 900 rpm, por um período de uma hora (GUILMORE, 2008).

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Implementar um sistema especialista de apoio a decisão na plataforma Arduino, utilizando a lógica *Fuzzy* para classificar a faixa da frequência cardíaca e recomendar a rotação ideal para a pedalada de praticantes de exercício físico em uma bicicleta ergométrica. Além disso, visa utilizar um gerador acoplado ao volante de inércia da bicicleta ergométrica para a conversão da energia mecânica produzida pelo usuário em energia elétrica.

1.3.2 Objetivos Específicos

Para alcançar o objetivo geral, faz-se necessário executar os seguintes objetivos específicos:

- Efetuar a revisão da literatura acerca do tema;
- Adaptar gerador CA no sistema de transmissão de força de uma bicicleta ergométrica (Laboratório de máquinas elétricas – DAELT-UTFPR-CT);
- Construir a placa de circuito impresso para promover a integração entre: sistema de sensoriamento, controle e atuação;
- Simular o sistema de apoio a decisão baseado em lógica *Fuzzy* no software MATLAB®; *Academic License R2018a (940813654) license number: 40685084*.
- Construir o código de funcionamento do sistema na plataforma de prototipagem eletrônica ARDUINO® Mega 2560;
- Coletar dados de: batimento cardíaco, carga e rotação das amostras;
- Apresentar resultados e conclusões por meio da comparação dos resultados obtidos: simulados e em laboratório (amostras).

1.4 JUSTIFICATIVA

O exercício físico regular representa um importante fator para diminuir índices de morbimortalidade cardiovascular, mas realizado de forma intensa, provoca variações no comportamento da frequência cardíaca (FC).

Pela facilidade de medição do seu comportamento a frequência cardíaca tem sido bastante estudada durante diferentes tipos e condições associadas ao exercício, pois é um dos meios usados para prescrever e controlar a intensidade de esforço físico (ALMEIDA E ARAÚJO, 2000).

A observação e o acompanhamento frequente da frequência cardíaca propicia alguns benefícios como (MARINS et, al 1998):

- acompanhar a intensidade do exercício físico;
- monitorar e supervisionar os efeitos do exercício físico no sistema cardiovascular;
- possibilitar a criação de planos de treinos para atingir objetivos com o exercício por exemplo perder peso, melhorar condicionamento;
- verificar se o praticante de exercício apresenta uma condição de cansaço físico.

Neste cenário, o ciclismo é uma atividade física aeróbica, e devem ser controlados os aspectos físicos dos praticantes para que não haja problemas de saúde durante a atividade.

Dado que, o ciclismo é considerado uma das formas mais eficientes para se promover a conversão do esforço humano de energia mecânica em elétrica é viável instalar geradores nos volantes de inércia das bicicletas ergométricas para efetuar geração de energia a partir do movimento de pedalada, podendo esta energia ser aplicada na alimentação das instalações de academias e centros de esportes, reduzindo o custo de energia elétrica.

Grandes quantidades de energia, proveniente do esforço humano, são geradas todos os dias com as atividades esportivas e recreativas, sendo que a recuperação dessa energia dissipada (mecânica) tenha sido foco de vários estudos. As atividades como: ciclismo, levantamento de peso, caminhada e corrida têm sido pesquisadas para quantificar a energia mecânica convertida em elétrica, em cada uma das modalidades, para posteriormente definir a sua empregabilidade no reuso (GUILMORE, 2008).

Atualmente, a geração de energia elétrica pelo esforço humano é justificada pela crescente demanda por recursos energéticos para sustentação do modo de vida populacional, aliada a escassez de recursos naturais (empregados na geração dos tipos de energia), que resulta na necessidade de desenvolvimento de fontes de energia renováveis.

Neste contexto, o presente trabalho justifica-se pelo desenvolvimento de um sistema para geração de energia elétrica de forma sustentável (aplicação de esforço em uma bicicleta ergométrica) conjugado com sistema de apoio e monitoramento dos batimentos cardíacos dos praticantes, apontando a intensidade ideal do exercício para promover prevenção de anomalias cardíacas.

1.5 ESTRUTURA DO TRABALHO

Este trabalho está organizado em seis capítulos, com as seguintes estruturas de tópicos:

Capítulo 1: é apresentada a introdução, assim como os objetivos, justificativa e estrutura do trabalho;

Capítulo 2: refere-se à revisão de literatura abordando os principais conceitos teóricos sobre: Fontes Renováveis de Energia, Geração de Energia via esforço Humano, Gerador de Energia, Lógica *Fuzzy* e um breve estudo sobre frequência cardíaca;

Capítulo 3: contempla-se a classificação da metodologia aplicada para o desenvolvimento da pesquisa e o detalhamento do protocolo de pesquisa;

Capítulo 4: este capítulo apresenta o desenvolvimento do sistema de geração de energia: acoplamento do gerador de energia na bicicleta ergométrica (disponível no laboratório de eletrotécnica da UTFPR-CT), construção da placa de circuito impresso, simulação do sistema no software MATLAB[®], detalhamento da programação executada no microcontrolador ARDUINO[®] e coleta de dados da realização de testes na bicicleta ergométrica.

Capítulo 5: Descreve os resultados e discussão acerca da comparação da simulação e testes executados em laboratório;

Capítulo 6: Descreve as conclusões e recomendações para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo foram descritos alguns conceitos básicos para geração de energia elétrica por microgeração de energia via esforço humano, gerador síncrono, lógica *Fuzzy* e uma breve revisão sobre frequência cardíaca. Esses fundamentos teóricos são de fundamental importância para uma melhor compreensão do princípio de funcionamento e das características do sistema desenvolvido.

2.1 MICROGERAÇÃO DE ENERGIA ELÉTRICA PELA PRÁTICA DE EXERCÍCIOS FÍSICOS

A Resolução Normativa RN nº 482 (17 de Abril de 2012), que estabelece as condições gerais para o acesso de microgeração distribuída aos sistemas de distribuição de energia elétrica, define que a microgeração distribuída é uma central geradora de energia elétrica, com potência instalada menor ou igual a 75 kW, e que utilize cogeração qualificada ou utilize fontes renováveis de energia elétrica, conectada na rede de distribuição por meio de instalações de unidades consumidoras.

A RN nº 482 - Agência Nacional de Energia Elétrica (ANEEL), possibilitou um desenvolvimento no número significativo de microgeradores, os quais ao serem conectados na rede de distribuição da concessionária, promovem a redução do custo de energia do usuário (DIAS; LIMA; RODRIGUES, 2016).

As normativas da ANEEL relacionadas à microgeração de energia elétrica, simplificam os processos de geração de energia pelos próprios consumidores, com acesso ao sistema de distribuição e as unidades consumidoras com microgeração (DIAS; LIMA; RODRIGUES, 2016).

Uma forma de geração de energia elétrica caracterizada por microgeração, em pequena escala, é a geração de energia por meio de esforços físicos humanos, a qual pode ser transformada em energia elétrica e armazenada por meio de baterias, e/ou utilizada na alimentação de instalações de centros esportivos.

A definição de trabalho mecânico pela lei da física é a capacidade de produzir energia, e essa energia ser transferida para um sistema, pela aplicação de uma força ou momento de força sobre ele, ao longo de um determinado percurso ou movimento de rotação (CALHEIROS, 2016).

A energia mecânica gasta por seres humanos quando executam atividades como: caminhada, esteira, elíptico entre outros, tem sido desperdiçada no que diz respeito a utilizá-la como fonte de energia (CALHEIROS, 2016).

Em todos os lugares, diariamente, milhões de pessoas andam, correm, saltam e essa energia poderia ser utilizada com a sua conversão de mecânica para elétrica, a qual pode ser direcionada na alimentação de equipamentos e instalações (CALHEIROS, 2016).

Adaptar os equipamentos destinados à prática de esportes para produzirem energia elétrica e fazê-la alimentar a infraestrutura do próprio ginásio como, sistemas de iluminação e equipamentos, transforma o sistema energético das academias em um ciclo sustentável (CHALERMTHAI et al., 2015).

Um dos equipamentos que tem sido adaptados para gerar energia elétrica é a esteira motorizada, a qual faz conversão de energia mecânica em energia elétrica. A energia mecânica produzida pela correia no tapete rolante é transferida para um gerador que fornece energia elétrica para carregar uma bateria. Outros equipamentos, são as máquinas elípticas, que em um treino de 30 minutos produz, em média, 50 Watts-hora de energia elétrica (CHALERMTHAI et al., 2015).

Outro exemplo são os pisos com cristais piezoelétricos, de 5mm, os quais são instalados no chão para conversão da pressão dos passos dos frequentadores do local em acumulação de carga elétrica. A diferença de potencial criada é utilizada para alimentar diretamente uma carga ou armazenar em uma bateria (CHALERMTHAI et al., 2015).

Em uma bicicleta ergométrica, para obter energia elétrica, acopla-se no volante inércia da bicicleta ergométrica um gerador, cujo o desempenho da energia gerada depende da experiência do usuário, ritmo e velocidade (cadência). Este sistema de microgeração, normalmente possui conversores nas suas saídas de energia disponibilizando uma tensão adequada aos equipamentos eletrônicos alimentados (CHALERMTHAI et al., 2015).

Em seguida apresentar-se-á uma breve descrição e algumas características do gerador síncrono utilizado na bicicleta ergométrica.

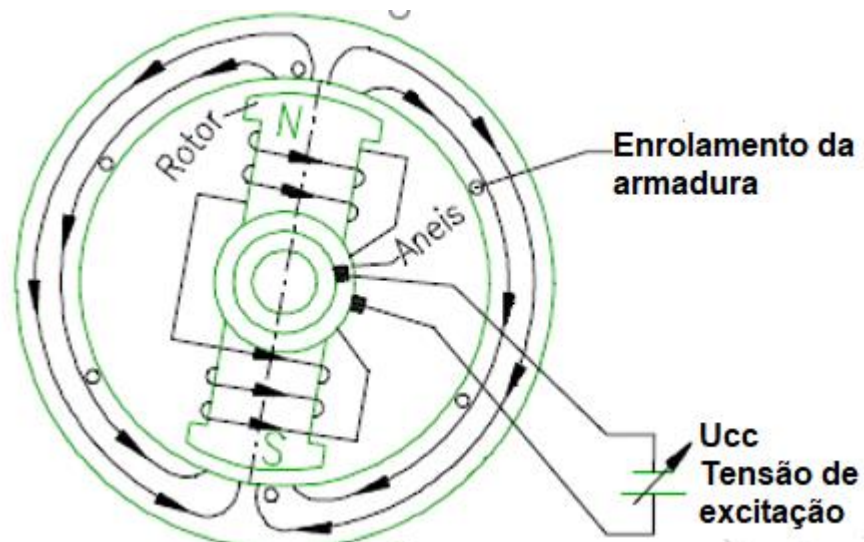
2.2 GERADOR SÍNCRONO

O gerador síncrono é utilizado pelas concessionárias de energia elétrica para fornecer potência trifásica e monofásica aos seus consumidores (DEL TORO, 1994).

A função principal de um gerador é de transformar a energia mecânica em elétrica. O princípio de funcionamento é um dos fundamentos básicos do eletromagnetismo em que uma bobina movimentada dentro de um campo magnético ou ao inverso, onde o elemento excitador, que produz o campo magnético, gira e extrai a tensão induzida nas bobinas da parte fixa do estator da máquina (NASCIMENTO, 2007).

Desta maneira, o rotor bobinado gira dentro de um campo eletromagnético que é produzido pela excitação de bobinas instaladas no estator da máquina ou aplicando-se uma tensão de excitação contínua ao rotor e este ao girar faz com que o campo magnético produzido nele, corte as bobinas do estator, resultando na criação de tensão induzida nessas bobinas. Para efetuar a conexão com a parte girante do gerador ao meio externo são conectados ao eixo do rotor dois anéis fixados que estão ligados ao meio externo por escovas. A figura 1 demonstra os aspectos construtivos do gerador síncrono (NASCIMENTO, 2007).

Figura 1- Aspectos construtivos do gerador síncrono



Fonte: Catálogo de Geradores Síncronos WEG MOTORS, 2003.

Quando o rotor gira sob a ação de uma força externa com velocidade constante dentro de um campo magnético B também uniforme e se v é a velocidade

linear do condutor em relação ao campo magnético segundo a lei de indução de Faraday, o valor da força eletromotriz induzida no condutor em movimento de rotação é definida por (MACAGNAN; NETO; NETO, 2016):

$$e = B \cdot l \cdot v \cdot \text{sen} \alpha \quad (1)$$

Onde: e = força eletromotriz gerada (volts);

B = campo magnético constante no tempo de indução (T);

l = comprimento do condutor que corta as linhas de campo (m);

v = velocidade linear do condutor (m/s);

$\text{sen} \alpha$ = seno do ângulo entre a velocidade do condutor e as linhas de campo magnético.

Desse modo, percebe-se que a medida em que o rotor gira o fluxo magnético concatenado varia senoidalmente entre os eixos magnéticos das bobinas do estator (defasadas de 120° geométricos) e do rotor.

Conforme Fitzgerald, (2006, p. 239 apud Macagnan, Neto, Neto, 2016) o nome síncrono é usado quando o rotor gira em sincronismo com o campo magnético girante produzido pelas correntes de armadura. A relação entre estas grandezas, pode ser verificado pela Equação (2).

$$n_s = \frac{120 \cdot f}{p} \quad (2)$$

onde: n_s = velocidade síncrona (rpm);

f = frequência (Hz);

p = número de polos;

Seguidamente apresentar-se-ão as características do gerador síncrono de ímãs permanentes.

2.2.1 Gerador síncrono de imã permanente

Dentre as máquinas síncronas, são destacadas as máquinas de imã permanentes, pois estas máquinas possuem um elevado desempenho e não dispõem de anéis coletores e nem enrolamentos de campo porque esses componentes são substituídos por elementos magnéticos que são os imãs permanentes.

De acordo com Nascimento Junior (2007, p. 225), o gerador síncrono de imã permanente é composto de um rotor que é um imã permanente de alta qualidade e um estator de aço silício laminado, como resultado a corrente alternada será gerada diretamente nas bobinas do estator, dispensando assim a necessidade de anéis e escovas.

O seu princípio de funcionamento assemelha-se com o gerador convencional, pois é baseado na excitação simultânea das fases para obtenção do campo girante (MACAGNAN; NETO; NETO,2016).

No contexto do sistema de geração de energia via pedais da bicicleta ergométrica, o gerador síncrono de imã permanente é o gerador adequado pois não necessita de alimentação externa para se transferir energia ao gerador. Quando o rotor deste gerador é posto em movimento de rotação por uma fonte primária, haverá uma variação no fluxo magnético que corta as bobinas dos enrolamentos da armadura, localizadas no estator, que irá gerar uma tensão induzida nas bobinas do estator.

Na sequência será mostrado os conceitos de lógica nebulosa e dos controladores *Fuzzy* usados para a elaboração deste trabalho.

2.3 LÓGICA *FUZZY*

A lógica *Fuzzy*, a qual foi desenvolvida em 1965 por Lotfi Zadeh (professor da Universidade da Califórnia, EUA), é uma forma lógica de raciocínio derivada da teoria dos conjuntos difusos para lidar com raciocínio aproximado (KUMRU; KUMRU 2013).

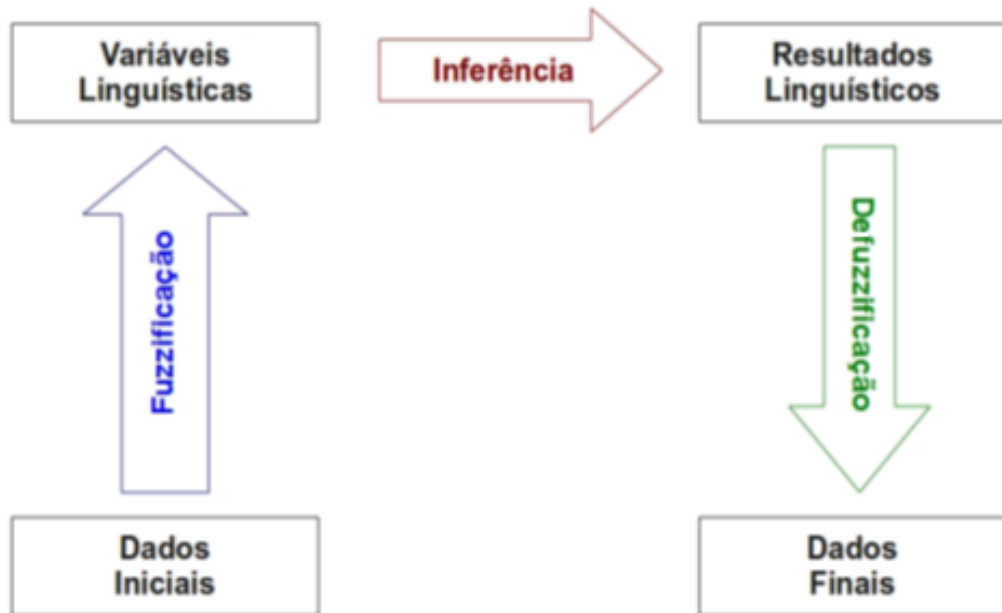
Tem como base a forma de como o cérebro trata as informações inexatas, sendo uma maneira de gerenciar as incertezas por meio da expressão de termos com um grau de incerteza num intervalo numérico de $[0,1]$ e faz o tratamento de expressões que contém grandezas de forma não exata (SHAW, SIMÕES, 2007).

A Lógica Nebulosa é fundamentada na teoria dos conjuntos *Fuzzy* sendo diferente dos sistemas lógicos tradicionais em suas características e seus detalhes. Nesta lógica, o raciocínio exato equivale a um caso limite do raciocínio aproximado, sendo interpretado como um processo de formação de relações nebulosas (GOMIDE; GUDWIN; TANSCHHEIT, 2012).

A teoria dos conjuntos nebulosos é usada no tratamento da natureza imprecisa que a informação pode assumir e a teoria dos conjuntos clássica é um caso particular desta teoria mais abrangente. Quando a teoria de conjuntos nebulosos é aplicada num contexto lógico, passa a ser chamada de lógica nebulosa, lógica difusa ou lógica *Fuzzy*. (SANDRI; CORREA, 1999).

Essa lógica utiliza de variáveis linguísticas (variáveis de entrada e saída), associadas com funções de pertinência para os processos de parametrização, fuzzificação e defuzzificação, sendo esta estrutura constituinte do processo de inferência *Fuzzy*. A Figura 2 mostra as etapas do processo de aplicação da metodologia *Fuzzy*.

Figura 2 – Sistema Lógico Fuzzy.



Fonte: Adaptado de Aguado e Cantanhede (2010 apud Cox,1994)

2.3.1 Conjuntos *Fuzzy*

Nesta seção serão apresentados os conceitos básicos sobre conjuntos e lógica *Fuzzy* cujo o objetivo é de adquirir conhecimento para o desenvolvimento do sistema de recomendação.

Um conjunto Nebuloso A do universo de discurso B é definido por uma função de pertinência $\mu_A: B \rightarrow [0,1]$. Essa função associa a cada elemento x de B o grau $\mu_A(x)$ com o qual x pertence a A (1).

Os conjuntos nebulosos definem as funções de pertinência, as quais expressam o quanto um dado elemento x pertence ao conjunto nebuloso A . A indica o grau de compatibilidade entre x e o conceito expresso por (NASCIMENTO JUNIOR; YONEYAMA, 2000; SANDRI, CORREA, 1999):

$\mu_A = 1$ indica que x é completamente compatível com A ;

$\mu_A = 0$ indica que x é completamente incompatível com A ;

$0 \leq \mu_A(x) \leq 1$ indica que x é parcialmente compatível com A , com grau $\mu_A(x)$.

A normatização da definição de conjuntos nebulosos pode ser obtida estendendo-se a teoria clássica de conjuntos . Um conjunto pode ser determinado pela sua função indicadora, dado um conjunto A no Universo B , defini-se $\mu_A: B \rightarrow \{0,1\}$ por :

$$\mu_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases} \quad (3)$$

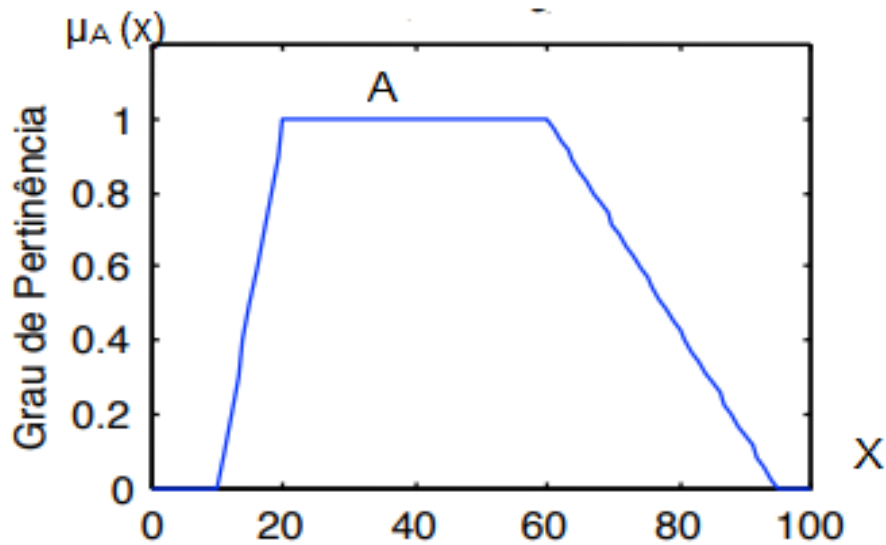
Os conjuntos nebulosos são determinados de acordo com a equação 4:

$$A = \{x \in B \mid \mu_A(x) = \varepsilon, 0 \leq \varepsilon \leq 1\} \quad (4)$$

Onde: $\mu_A(x) = B \rightarrow \{0,1\}$

A Figura 3 ilustra a função de pertinência estruturada pelos valores do conjunto nebuloso A .

Figura 3 - Função de Pertinência para um conjunto Nebuloso



Fonte: Nascimento, Junior e Yoneyama (2000).

Conforme o gráfico da Figura 3, a função de pertinência formada é tipo trapezoidal, a qual foi estruturada obtendo o grau de pertinência de x no conjunto nebuloso A . Uma função de pertinência é uma função numérica gráfica ou tabulada que dá valores de pertinência *Fuzzy* para valores discretos de uma variável, que pertencem ao Universo de discurso (SHAW; SIMÕES, 2007). Classifica-se como Universo de discurso o espaço completo de variação (range) de uma variável do modelo (SANDRI; CORREA, 1999).

As funções de pertinência representam os aspectos fundamentais de todas as ações teóricas e práticas de sistemas *Fuzzy*. Os formatos encontrados são: Linear, triangular, trapezoidal e Gaussiana, sendo que essas funções não precisam ser simétricas (SHAW; SIMÕES, 2007).

Desta forma, a álgebra entre os conjuntos *Fuzzy* não segue a teoria clássica normal, porque as operações de união e de intersecção de um conjunto e seu complementar são diferentes de 1 e vazio (NASCIMENTO JUNIOR; YONEYAMA, 2000).

Por sua vez, as funções de pertinência atribuem o grau de pertinência para os números *Fuzzy* transformando valores de entrada não-*Fuzzy* em termos linguísticos e vice-versa, ou seja, quantificam os termos linguísticos utilizando funções de pertinência. As funções de pertinências serão utilizadas no processo de fuzzificação

e defuzzificação da inferência *Fuzzy*, uma vez que os números *Fuzzy* mais utilizados pela metodologia proposta são: triangular e trapezoidal (KUMRO; KUMRO, 2005).

Logo, para realizar a conversão de um número *Fuzzy* $\tilde{A} = [x_1, x_2, x_3]$ em um conjunto A^α , utiliza-se a equação (5) (ARAÚJO, 2008).

$$A^\alpha = [(x_2 - x_1)\alpha + x_1, -(x_3 - x_2)\alpha + x_3] \quad (5)$$

Dessa forma, nessa conversão é gerado um conjunto $\alpha - cut$ que é uma restrição imposta ao domínio, baseada no valor de α . Os elementos do domínio que possuem $\mu(x)$ acima de um certo valor de α estão contidos nesse conjunto. Possuem dois elementos que representam o limite superior e inferior do intervalo do conjunto, (equação (4)), uma vez que os valores são encontrados substituindo α no intervalo $[0,1]$ (ARAÚJO, 2008).

$$A^\alpha = [x_1^\alpha, x_3^\alpha] \quad (6)$$

As funções de pertinência que atribuem grau de pertinência ($\mu_{\tilde{A}}$) para os números *Fuzzy* triangular e trapezoidal contidos na lógica *Fuzzy* são evidenciados no Quadro 1.

Quadro 1 – Função de Pertinência Utilizada na Lógica Fuzzy

Número <i>Fuzzy</i> Triangular	Número <i>Fuzzy</i> Trapezoidal
$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x - x_1}{x_2 - x_1}, & \text{se } x \in [x_1, x_2] \\ \frac{x_3 - x}{x_3 - x_2}, & \text{se } x \in [x_2, x_3] \\ 0 & \text{caso contrário} \end{cases}$	$\mu_{\tilde{A}}(x) = \begin{cases} 0, & \text{se } x < x_1; \\ \frac{x - x_1}{x_2 - x_1}, & \text{se } x \in [x_1, x_2] \\ 1, & \text{se } x \in [x_2, x_3] \\ \frac{x_4 - x}{x_4 - x_3}, & \text{se } x \in [x_3, x_4] \\ 1, & \text{se } x < x_4 \end{cases}$

Fonte: Adaptado de ARAÚJO (2008)

2.3.2 Operações em Conjuntos Nebulosos

Sejam A e B conjuntos nebulosos definidos em ε . Pode-se expressar a interseção desses conjuntos como $E = A \cap B$ e para expressar a união de um conjunto temos que $F = A \cup B$ (SANDRI; CORREA., 1999).

Na teoria dos conjuntos nebulosos, a interseção é realizada por uma família de operadores identificados de t – normas (operadores mínimos), e a união é implementada por uma de operadores identificados de s- normas (operadores máximos) (NASCIMENTO JUNIOR E YONEYAMA, 2000) e (GOMIDE, GUDWIN, TANSCHHEIT. 2012). A seguir serão descritos cada operador.

2.3.2.1 Operadores s-norma

- União de A e B: $\mu_{A \cup B}(x) = \mu_A(u) \cup \mu_B(u)$

Verifica-se que a união é o menor subconjunto do universo de discurso U, que contém ambos os conjuntos A e B. Como a união é o contorno que inclui ambos os conjuntos A e B, o resultado é sempre maior que os conjuntos individuais. Por este motivo, pode-se considerar o vetor de pertinência para a união $A \cup B$ (IVANQUI,2005).

Onde, s é a co-norma triangular de uma união coletiva. A norma-s é uma função de duas entradas delimitada nos domínios s: $[0,1] \times [0,1] \rightarrow [0,1]$, onde “X” representa o produto cartesiano. O operador -s indica um mapeamento entre duas funções de pertinência fuzzy, cada uma no intervalo $[0,1]$.Tal função deve satisfazer certas condições, tais como (NASCIMENTO JUNIOR;YONEYAMA, 2000;IVANQUI,2005):

$$\begin{aligned} \text{Condições de contorno: } u \text{ s } 0 &= u, \forall u \in [0,1] \\ u \text{ s } 1 &= 1, \forall u \in [0,1] \end{aligned}$$

Onde as seguintes propriedades e premissas, são válidas:

- Propriedade de contorno $s(1,1) = 1$;
- Propriedade comutativa: $s(a,b) = s(b,a)$;
- Propriedade associativa: $s(a,s(b,c)) = s(s(a,b),c)$;

- Condições monotônicas: para $a \leq c$ e $b \leq d \rightarrow s(a,b) \leq s(c,d)$;
- Propriedade de contorno $s(b,0) = b$;
- Intersecção de A e B: $\mu_{A \cap B}(x)$.

2.3.2.2 Operadores t-norma

A intersecção é o maior subconjunto do universo de discurso U, que é ao mesmo tempo parte de A e também de B, portanto, sempre menor que os conjuntos individuais de A e B (IVANQUI,2005).

t é a norma triangular ou norma-t de intersecção coletiva. A norma-t é uma função de duas entradas delimitada nos domínios $t: [0,1] \times [0,1] \rightarrow [0,1]$, onde "X" representa o produto cartesiano. O operador-t indica um mapeamento entre duas funções de pertinência *Fuzzy*, cada uma no intervalo $[0,1]$; a operação-t é produzida entre elas, de acordo com sua forma ponto a ponto e dita ser uma t-norma se (IVANQUI,2005):

- Propriedade de contorno $t(0,0) = 0$;
- Propriedade comutativa: $t(a,b) = t(b,a)$;
- Propriedade associativa: $t(a,t(b,c)) = t(t(a,b),c)$;
- Condições monotônicas: para $a \leq c$ e $b \leq d \rightarrow t(a,b) \leq t(c,d)$;
- Propriedade de contorno $t(a,1) = a$.

2.3.2.3 Propriedades Algébricas utilizando as definições de união e intersecção

De acordo com Nascimento e Yoneyama (2000) e Gomide, Gudwin, Tanscheit (2012) utilizando as definições de união e intersecção, efetuadas pelos operadores máximo e mínimo, e complemento, é provável verificar que as seguintes propriedades algébricas de conjuntos ordinários também valem para conjuntos *Fuzzy* (NASCIMENTO JUNIOR; YONEYAMA, 2000 ;IVANQUI,2005).

Propriedades Algébricas:

Involução: $(A^a)^a = A$

De Morgan: $(A \cap B)^c = A^c \cup B^c$

De Morgan: $(A \cup B)^c = A^c \cap B^c$

Idempotência: $A \cap A = A$ e $A \cup A = A$

Comutatividade: $A \cap B = B \cap A$ e $A \cup B = B \cup A$

Associatividade: $(A \cap B) \cap C = A \cap (B \cap C)$ e $(A \cup B) \cup C = A \cup (B \cup C)$

Absorção: $(A \cap B) \cup A = A$

Absorção: $(A \cup B) \cap A = A$

Distributividade: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ ou

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Lei Transitiva: se $A \subset B$ e $B \subset C$ então $A \subset C$

Considerando que as funções de pertinência dos conjuntos vazio e universo são determinadas como sendo 0 e 1, respectivamente, $\forall u \in U$, as seguintes propriedades também são válidas:

$A \cap \emptyset = \emptyset$ e $A \cap U = A$

$A \cup \emptyset = A$ e $A \cup U = U$

2.3.2.4 Variáveis Linguísticas

Uma variável linguística é uma variável cujos valores são nomes de conjuntos *Fuzzy*. como, o batimento cardíaco de um indivíduo é uma variável linguística quando assume os valores de frequência cardíaca mínima, frequência cardíaca ideal, frequência cardíaca máxima e de frequência cardíaca risco. Estes valores são descritos por intermédio de conjuntos *Fuzzy*, os valores de uma variável linguística podem ser sentenças em uma linguagem especificada (GOMIDE;GUDWIN; TANSCHHEIT, 2012).

Segundo Aguado e Cantanhede (2010 apud Von Altrock,1996 as variáveis linguísticas como sendo a nomenclatura da linguagem *Fuzzy*, porque são expressões presentes na linguagem humana e que quando traduzidas ou interpretadas podem levar a conclusões importantes. Perante este contexto na lógica *Fuzzy* a função de pertinência vista na seção anterior vem a ser um mapeamento matemático de cada valor numérico possível para as variáveis linguísticas.

2.3.2.5 Operadores de Implicação e proposições *Fuzzy*

De acordo com Gomide, Gudwin, Tanscheit (2012), uma proposição *Fuzzy* é uma representação de uma parte do conhecimento e é definida no formato (X é A) onde X é o nome de uma variável linguística e A é um subconjunto *Fuzzy* definido no universo de discurso U de X.

Proposições *Fuzzy* podem ser combinadas utilizando-se diferentes operadores, criando novas proposições *Fuzzy*, como exemplo tem-se operadores que contêm os conectivos lógicos E e OU, do mesmo modo que o operador de implicação Se, Então (GOMIDE; GUDWIN; TANSCHHEIT, 2012).

Na criação de uma máquina de inferência, é fundamental especificar quais são as t-normas e s-normas usadas para realizar as operações E e OU e para implicação e agregação de regras (IVANQUI,2005).

Os operadores de implicação $I = [0, 1]^2 \rightarrow [0, 1]$ são utilizados para formar as regras de inferência do tipo **Se** <premissa> **Então** <conclusão>. Considerando A e B dados por $\mu_A : \rightarrow X [0, 1]$, $\mu_B : \rightarrow Y [0, 1]$, a relação de $A \rightarrow B$ é apresentada como (SANDRI; CORREA., 1999):

$$\mu_{A \rightarrow B}(x,y) = I(\mu_A(x), \mu_B(y)) \quad (7)$$

De acordo com a definição a implicação é uma relação que é expressa por “Se X Então S”. Isto se estabelece a partir do conhecimento do especialista na forma de proposições do tipo “se X1 e/ou Y2 então S1”, onde X1 e Y2 referem-se a classificações linguísticas das variáveis de entrada e S1 a de saída, as quais, a partir do uso desta teoria, se mostra como consequência das operações “e/ou” algébricas acima referidas, onde “ou” designa união e “e” intersecção (GOMIDE; GUDWIN; TANSCHHEIT, 2012).

As regras do tipo Se-então são um tipo de combinação de proposições que dão origem às regras de inferência *Fuzzy*.

A teoria de emprego de conjuntos *Fuzzy*, do tipo Se <premissa> Então <conclusão> vai decidir as ações do controle, resultando assim em sistemas de decisão, estruturados no formato de uma base de regras *Fuzzys* (SANDRI; CORREA, 1999).

2.3.3 Sistema *Fuzzy*

Um sistema *Fuzzy* típico pode ser configurado para comportar-se de acordo com o raciocínio dedutivo, utilizando a técnica que os humanos usam para deduzir conclusões baseadas em informações que eles já conhecem. A lógica *Fuzzy* pode usar a estratégia que as pessoas utilizam para dedução em um controlador *Fuzzy*, viabilizando, assim, a construção de um controlador computacional com processamento de dedução semelhante ao humano (SHAW; SIMÕES, 2007).

“O raciocínio dedutivo é uma maneira diferente, que também pode ser utilizado no projeto de controladores *Fuzzy*, onde seria possível o aprendizado e generalização através de exemplos particulares provenientes da observação do comportamento do processo numa situação dinâmica, ou variante no tempo” (SHAW; SIMÕES, 2007).

A intenção do controle *Fuzzy* é modelar as ações a partir de conhecimento especialista, e não de modelar o processo em si. Ao contrário dos métodos convencionais de controle de processos, onde são modelados por ferramentas matemáticas (GOMIDE; GUDWIN; TANSCHKEIT, 2012).

A modelagem e o controle *Fuzzy* são técnicas para se manejar informações qualitativas de uma maneira mais correta possível, considerando o jeito como a falta de exatidão e a incerteza são descritas, tornando-se ferramentas poderosas para compor de maneira conveniente o conhecimento (LEE, 1990).

O sistema *Fuzzy* cuida da relação entre entradas e saídas, associando vários parâmetros de processo e de controle, resultando em um processo mais aperfeiçoado, com desempenho estável e robusto. Sua maneira singela de implementação de sistemas de controle *Fuzzy* pode reduzir a complexidade de um projeto (GOMIDE; GUDWIN; TANSCHKEIT, 2012).

De acordo com Tavares (2009), o controle *Fuzzy* tem os sinais de entrada obtidos através de sensores ou outro sistema de medição que é submetido a um tratamento adequado do sinal analógico o qual fará a alimentação do bloco de fuzificação (*Fuzzy sets*) do controlador *Fuzzy*. A base de conhecimento deste controle é armazenada por meio de um conjunto de regras disparadas através do método de inferência cujas decisões são tomadas e os consequentes das regras *Fuzzy* resultantes são devolvidas ao sistema ainda em linguagem *Fuzzy*. O bloco de

defuzzificação faz a conversão em informações no formato “*crisp*” que é atribuído na saída do controlador.

Para que o sistema atue de um modo aceitável, que neste contexto é de monitorar a frequência cardíaca e de recomendar a rotação adequada de quem está pedalando, as variáveis de entrada do sistema são 3 pessoas aplicando esforço ao pedal, cujas variações da potência mecânica e da rotação são de acordo com a experiência, ritmo e velocidade do usuário e a variável de saída do sistema de recomendação *Fuzzy* é a rotação adequada à atividade física.

Na elaboração de cada regra deve-se definir a conexão entre as variáveis de entrada e as variáveis de saída, através dos operadores lógicos. Após a inserção de todas as regras tem-se a formação do sistema de controle *Fuzzy*.

A seguir será tratada a configuração básica de um sistema *Fuzzy*.

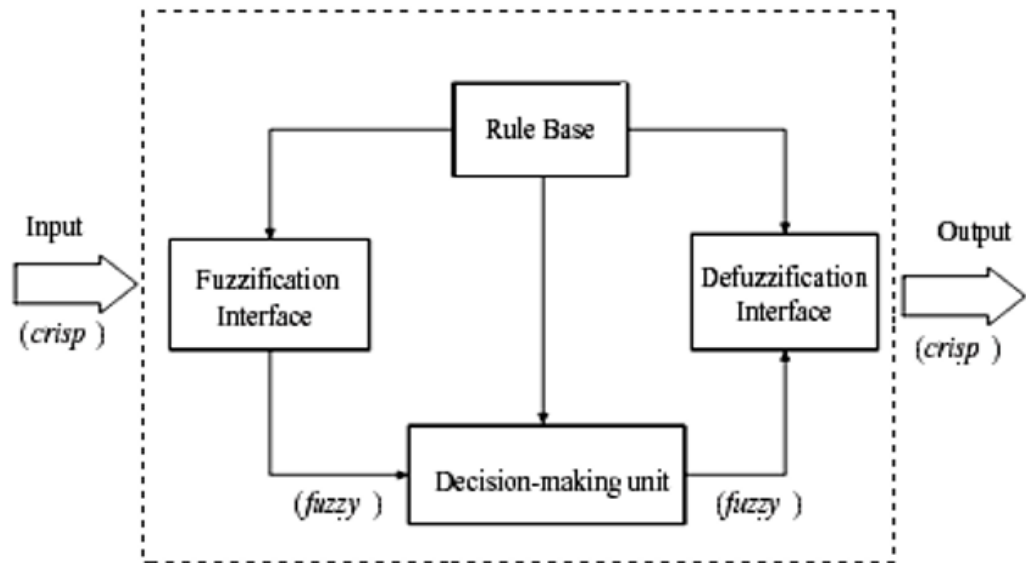
2.3.3.1 Estrutura de um sistema *Fuzzy*

De acordo com Shaw e Simões (2007), um sistema *Fuzzy* é composto dos seguintes blocos funcionais:

- Fuzzificação;
- Base de conhecimento;
- Regras de tomadas de decisões;
- Defuzzificação.

A Figura 4 mostra a configuração básica de um sistema *Fuzzy*:

Figura 4- Configuração básica de um Sistema Fuzzy.



Fonte: Robotic Research Group (RRG)- Universidade Estadual do Piauí UESPI-
Teresina, 2012.

Shaw, Simões (2007) e Andrade, Jacques (2008) definem os blocos funcionais (Figura 4) como:

- Fuzzificação: Nesta etapa, o problema é investigado e os dados de entrada são transformados em variáveis linguísticas, e o próximo passo é determinar as funções de pertinência que podem ser um conjunto tabulado de valores numéricos.

- Base de conhecimento: consiste de uma base de regras, caracterizando a estratégia de controle e suas metas. Armazena as definições necessárias sobre discretizações e normalizações dos universos de discurso. As proposições são geradas da união entre as variáveis do modelo e a região Fuzzy. Essas regras resultantes das associações podem ser condicionais ou não condicionais sendo embasadas nas funções de pertinência e uma base de regras Fuzzy linguísticas que é a estratégia de controle utilizada.

- Regras de lógica de tomada de decisão: processa os dados fuzzy de entrada, junto com as regras. Utiliza inferência Fuzzy para simular tomadas de decisão humanas gerando ações consequentes a partir de um conjunto de condições de entrada antecedentes.

- Defuzzificação: é a etapa em que os valores Fuzzy são convertidos em números reais, tendo, assim, um conjunto de saída matematicamente definido, sendo o valor da variável linguística de saída inferida interpretada num valor discreto.

- O método de defuzzificação utilizado neste trabalho é o Centro de Área (COA) ou de gravidade, nesse o valor de saída é o centro de gravidade da função de distribuição retornando assim, o centro de área da função inferida (ANDRADE; JACQUES, 2008; OLIVEIRA; NASCIMENTO, 2008).

- A metodologia de programação para controle Fuzzy utilizado nesse trabalho foi o estudo de MAMDANI realizado em 1973. Neste controlador de MAMDANI a elaboração de cada regra é feita por meio da determinação de operadores para o processamento do antecedente da regra e da função de implicação que irá determinar o seu consequente (ANDRADE; JACQUES, 2008).

- A ação do controlador Fuzzy é definida pela associação das regras implementadas que compõem o algoritmo, por meio da utilização do conectivo “também”, o qual pode ser processado por diferentes operadores. Esta associação resulta no conjunto Fuzzy C, que define a saída do controlador C. A saída efetiva do controlador é, então, obtida por meio de um processo de defuzzificação aplicado ao conjunto C, que nesse trabalho é o centro de área (ANDRADE; JACQUES, 2008).

A seguir será apresentado o dispositivo para efetuar o controle do sistema.

2.4 MEDIÇÃO DA FREQUÊNCIA CARDÍACA

Para Caputo, Silva e Rombaldi (2012) e Eleno e Kokubun (2002), uma das variáveis fisiológicas mais usadas para indicação de cargas de trabalho para planos de exercícios é a frequência cardíaca máxima (FCMáx) pois ela está diretamente ligada ao esforço imposto pelo coração. A frequência cardíaca sofre influência da intensidade dos exercícios físicos e tem a tendência de aumentar quando é atribuído um aumento gradual da carga de trabalho.

Muitas equações podem ser utilizadas para determinar a frequência cardíaca máxima do ser humano, e expressa em batimentos por minuto (bpm). No entanto, a mais usual é a equação de Karnoven, na qual:

$$FC \text{ máx} = 220 - \text{Idade} \quad (8)$$

Durante a pesquisa adotou-se esta equação devido a ser aceita pelo *American College of Sports Medicine*, pois o FC máx (de acordo com o método de Karnoven) é em função da idade com batimento cardíaco máximo de 220 bpm (batimento por minuto) (IVANQUI, 2005; CAPUTO; SILVA; ROMBALDI; 2012).

De acordo com Karnoven et al. ¹ (1957 apud Ivanqui, 2005), a resposta da equação 8 deve ser subtraído de uma margem de cobertura, de + ou - 10 para pessoas com idade inferior a 25 anos e a partir de 25 permite-se uma abrangência maior de + ou - 12

A metodologia de Karvonen, cita que um indivíduo necessita fazer o monitoramento de seus batimentos cardíacos entre a faixa mínima e máxima durante a atividade física, condicionando seu treinamento nos valores determinados pelas equações presentes no Quadro 2 (IVANQUI, 2005):

Quadro 2 - Classificação das Frequências Cardíacas: Mínima, Ideal e Máxima

Classificação da frequência Cardíaca	Equação
Frequência cardíaca mínima	FCM x 0,60 (9)
Frequência ideal na atividade aeróbica	FCM x 0,70 (10)
Frequência cardíaca máxima	FCM x 0,85 (11)

Fonte: adaptado de IVANQUI (2005).

A potência mecânica medida em watts (W) é uma das variáveis fisiológicas de treinamento usadas pelos ciclos ergômetros, pois há uma relação entre potência mecânica e cadência (DIAS; LIMA; NOVAES, 2016).

De acordo com os estudos de Caputo (2012), a cadência ideal está no intervalo de 60 até 90 rotações por minuto devido ao aumento da potência mecânica. Já acima de 90 rotações por minuto (rpm) ocorre diminuição da potência mecânica, diminuindo a eficiência.

No próximo capítulo será apresentada a metodologia e o protocolo de pesquisa utilizados nesse trabalho.

¹ KARVONEN, M.J; KENTAL, E.; MUSTALA, O. The effects of on heart rate a longitudinal study. *Ann. Med. Exper. Fenn.*, v. 35, p. 307-315, 1957.

3 MATERIAIS E MÉTODOS

3.1 CLASSIFICAÇÃO DA METODOLOGIA

De acordo com Lakatos e Marconi (2010), considera-se pesquisa como uma técnica formal que requer tratamento científico, e sua constituição é um meio para se conhecer a realidade, buscando solucionar questões sugeridas e utilizando como ferramenta métodos científicos.

A pesquisa é um procedimento intelectual, cuja meta do pesquisador é de obter conhecimento a partir da investigação de uma situação real e da procura de novas realidades sobre um problema. Com apoio de uma metodologia adequada o pesquisador procura o conhecimento específico para o problema estudado (FACHIN, 2005; SILVA; MENEZES, 2005).

A pesquisa aplicada utiliza-se de toda informação disponível para a criação de novas tecnologias e métodos, assim, esta pesquisa caracteriza-se como pesquisa aplicada, pois desenvolveu um sistema de monitoramento de batimento cardíaco e de carga utilizada na prática de exercícios físicos em um a bicicleta ergométrica agregado a um sistema de geração de energia elétrica de forma sustentável (PRODANOV; DE FREITAS, 2013).

Do ponto de vista da natureza do presente trabalho científico, esta pesquisa pode ser classificada como pesquisa aplicada, pois o conhecimento gerado pelo seu resultado pode ser aplicado em ambientes de academias e centros esportivos para a solução de problemas específicos relacionados à eficiência energética.

Segundo os procedimentos técnicos é uma pesquisa bibliográfica e documental, porque utilizou-se de livros, artigos, manuais e dissertações para a compreensão do sistema, oferecendo dados para a resolução do problema, ou seja, documentos já elaborados (PRODANOV; DE FREITAS, 2013).

Para Gil (2002) “quanto aos objetivos, esta pesquisa classifica-se como exploratória, pois visa proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses”. Logo, esta pesquisa se classifica, no objetivo, como exploratório, pois analisa os dados obtidos nos testes de esforços físicos e os valores obtidos na simulação efetuada no MATLAB®, com a finalidade de estabelecer a faixa adequada de batimento cardíaco e rotação durante a prática do exercício físico, para prevenir anomalias cardíacas.

Pesquisa quantitativa é centrada na objetividade, pois considera que a realidade só pode ser compreendida com base na análise de dados numéricos recolhidos com auxílio de instrumentos padronizados e outros (SILVA; MENEZES, 2005; LAKATOS; MARCONI, 2010)

Do ponto de vista da forma de abordagem do problema esta pesquisa classifica-se como quantitativa, pois compara dados quantitativos relacionados a simulação e testes reais em laboratório, da aplicação de esforço físico em uma bicicleta ergométrica, analisando dados relacionados à idade, batimento cardíaco, carga, rotação e potência gerada.

A classificação da metodologia aplicada no trabalho está representada no Quadro 3.

Quadro 3 – Classificação da metodologia (resumo)

QUESITO	CLASSIFICAÇÃO						
	Aplicada	Básica					
Natureza	Aplicada	Básica					
Objetivos	Exploratória	Descritiva	Explicativa				
Procedimentos	Pesquisa Bibliográfica	Pesquisa Documental	Pesquisas Experimental	Levantamento de dados	Observação Direta	Pesquisa de Campo	Estudo de caso
Abordagem	Pesquisa Quantitativa	Pesquisa Qualitativa					

- Utilizado - não utilizado

Fonte: autoria própria.

3.2 PROTOCOLO DE PESQUISA

A preparação do protocolo de pesquisa determina o modo de tratamento dos dados e informações obtidas e a estrutura do desenvolvimento do trabalho. O protocolo de pesquisa define a conduta adotada para a organização do estudo, conduzindo o pesquisador rumo ao conhecimento, desde sua formulação até o alcance dos objetivos que é de apresentar resultados.

O presente trabalho é estruturado nas seguintes etapas do protocolo de pesquisa: preparação da pesquisa, pesquisa aplicada, processamento de dados, análise de resultados e compartilhamento. Estas etapas estão detalhadas a seguir:

Preparação da pesquisa: nesta etapa é definido o problema da pesquisa, efetuada a revisão da literatura acerca do tema e adaptação do gerador CA no sistema de transmissão de força de uma bicicleta ergométrica (Laboratório de máquinas elétricas – DAELT-UTFPR-CT) para aplicação dos testes de esforço físico em função da geração de energia.

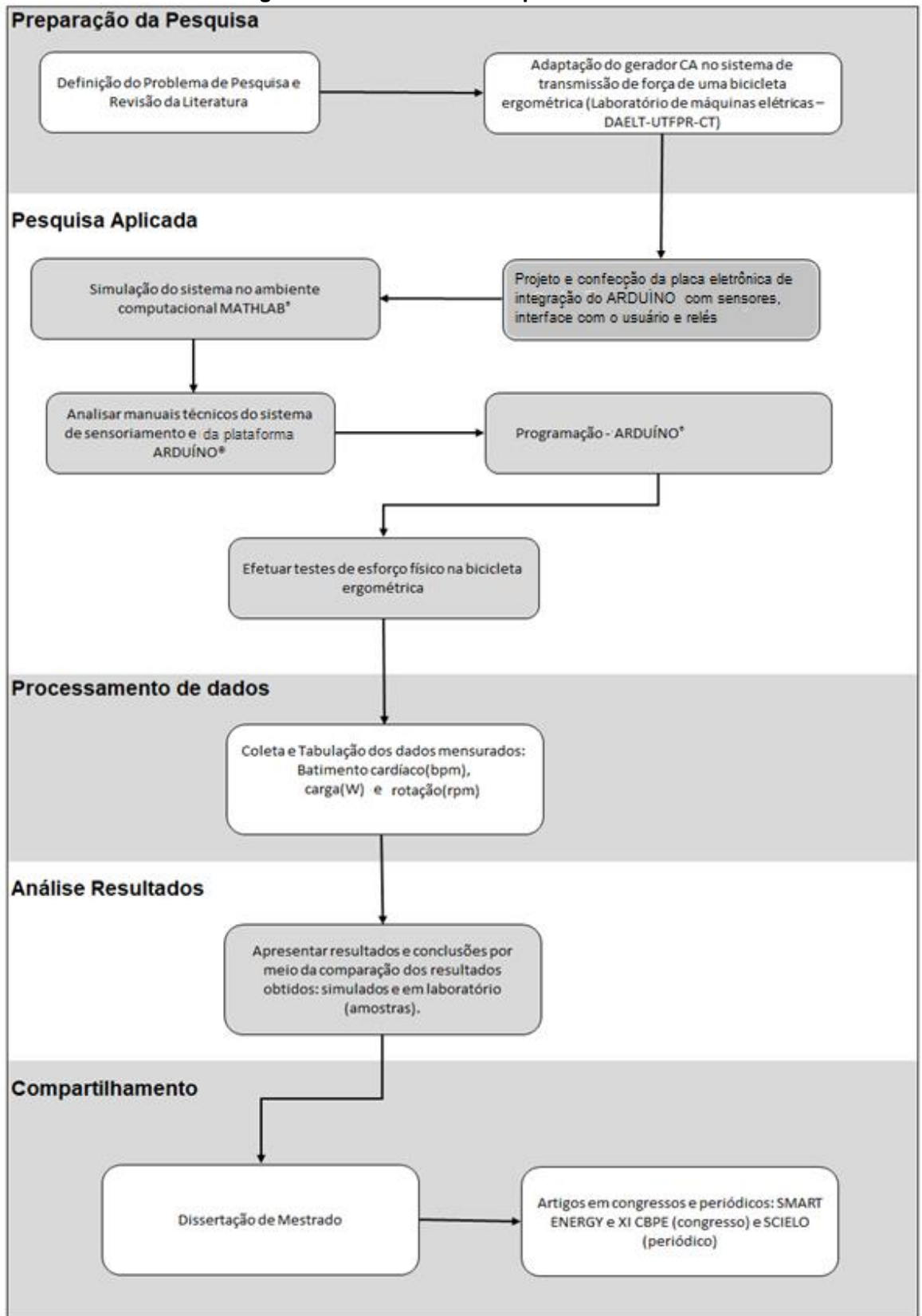
Pesquisa Aplicada: Simular no sistema no ambiente computacional MATLAB® para verificar se as respostas de saída estão de acordo com a programação pré-estabelecida. Projeto e confecção da placa eletrônica que fará a integração dos sensores, interface com o usuário e relés com a plataforma ARDUINO®. Analisar manuais técnicos do sistema de sensoriamento e ARDUINO®. Programação na plataforma ARDUINO® para construção do sistema de apoio a decisão (monitoramento e orientação); Efetuar testes de esforço físico na bicicleta ergométrica para verificar: o batimento cardíaco médio mensurado, a carga utilizada pelo praticante e rotação da bicicleta.

Processamento de dados: estruturar os dados coletados com o teste de esforço físico realizado em Laboratório e da simulação executada.

Análise de resultados: nesta fase descreve-se os resultados obtidos por meio da comparação dos parâmetros simulados e testes físicos realizados na bicicleta ergométrica do laboratório de eletrotécnica.

A Figura 5 ilustra a sequência do protocolo de pesquisa.

Figura 5 - Protocolo de Pesquisa



Fonte: autoria própria.

Compartilhamento: estrutura escrita da dissertação e submissão de artigos nos congressos SMART ENERGY (“Geração de energia elétrica obtida através de um gerador de corrente contínua acoplado ao pedal de uma bicicleta ergométrica”), XI CPBE (Microgeração de energia elétrica obtida através de um alternador acoplado ao pedal de uma bicicleta ergométrica com potência controlada por Lógica *Fuzzy*) e na base de consulta SCIELO(“Geração de energia elétrica obtida através de um gerador de corrente contínua acoplado ao pedal de uma bicicleta ergométrica”).

4 DESENVOLVIMENTO DO TRABALHO

4.1 SISTEMA DE APOIO À DECISÃO E DE GERAÇÃO DE ENERGIA SUSTENTÁVEL

Este trabalho refere-se ao desenvolvimento de um sistema de recomendação, baseado em lógica *Fuzzy*, que monitora os valores de batimento cardíaco e orienta a rotação adequada para o praticante, evitando anomalias cardíacas durante o exercício físico.

Agregado a este sistema de recomendação, foi construído um sistema de microgeração de energia sustentável obtida pela conversão da energia mecânica, a qual é gasta pelas pessoas durante a prática dos exercícios na academia, em energia elétrica, que pode ser utilizada na alimentação de demais equipamentos e instalações.

Para desenvolvimento deste sistema, seguiu-se as etapas:

- 1 - Adaptação do gerador CA no sistema de transmissão de força de uma bicicleta ergométrica (Laboratório de máquinas elétricas – DAELT-UTFPR-CT);
- 2 - Construção da placa de circuito impresso para promover a integração entre: sistema de sensoriamento, controle e interface com o usuário;
- 3 - Simulação do sistema de recomendação no software MATLAB®;
- 4 - Construção do código de funcionamento do sistema na plataforma ARDUINO®;
- 5 - Coleta dos dados de: batimento cardíaco, carga e rotação dos 3 eventos;
A seguir detalhar-se-ão as etapas de desenvolvimento deste trabalho.

4.2 ADAPTAÇÃO DO GERADOR (CA)

O sistema de microgeração de energia sustentável é composto de uma bicicleta ergométrica em estrutura metálica, no qual o pedal é fixado a uma polia. Para a microgeração de energia, um gerador CA foi acoplado no volante de inércia da polia maior, por um elemento de transmissão de movimento (polia). Também foi instalado na bicicleta um sistema de recomendação baseado em lógica *Fuzzy* desenvolvido a partir da plataforma ARDUINO® e placas de circuito impresso.

A Figura 6 mostra a bicicleta ergométrica com os aparatos acoplados.

Figura 6 - Bicicleta Ergométrica com Sistema de Recomendação e de Microgeração de energia

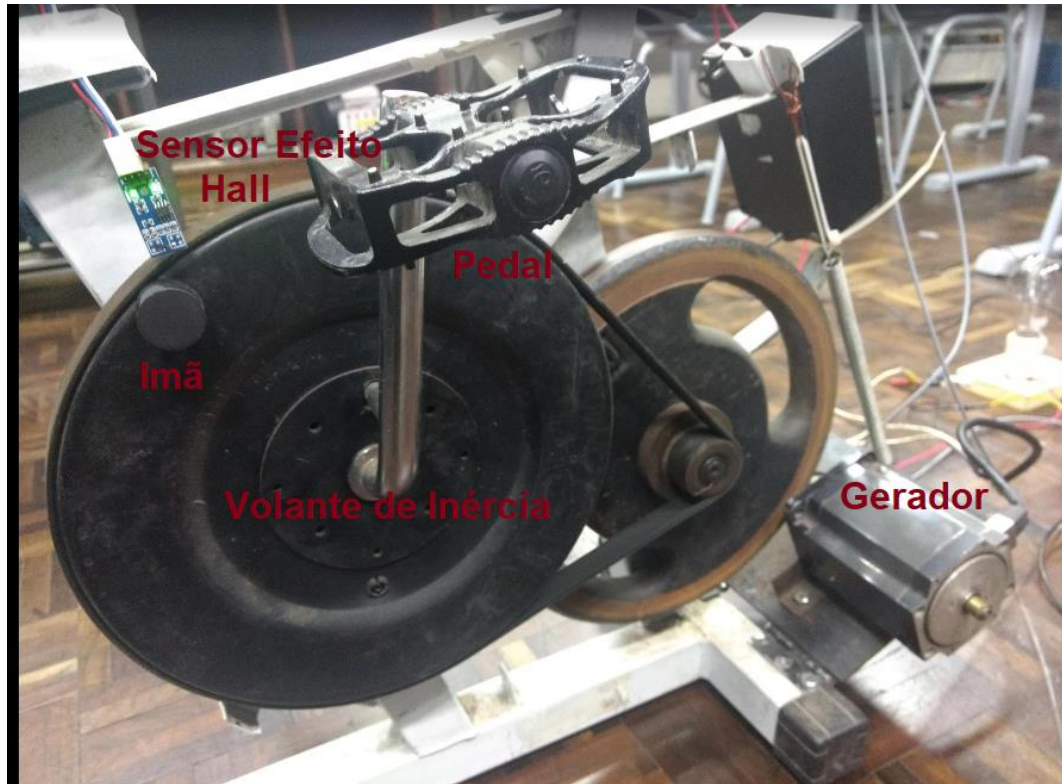


Fonte: autoria própria.

Os aparatos identificados na Figura 6 são: (1) Gerador CA; (2) Polia interligada ao pedal; (3) Pedal e (4) Placa do Sistema de Recomendação (fundamentado em lógica *Fuzzy*).

A Figura 7 mostra com melhor resolução o sistema de transmissão de movimento composto por polias e correias.

Figura 7- Sistema de Transmissão de Movimento e Geração



Fonte: autoria própria.

Este sistema de polias e correia interliga o pedal ao volante de inércia, o qual aciona o gerador por meio de movimento rotacional.

A máquina elétrica (gerador síncrono de ímã permanente) instalada na bicicleta ergométrica, foi concedida pelo departamento acadêmico de Eletrotécnica (DAELT), sendo ela fabricada pela Sanyo. A Tabela 1 mostra os valores nominais da máquina síncrona de corrente alternada:

Tabela 1 - Valores nominais do gerador síncrono AC

Potência (W)	Tensão (V)	Torque (N.m)	Corrente (A)	Rotação(rpm)
750	117	2,35	5,2	3000

Fonte: Dados Técnicos da Placa do Gerador fornecidos pelo fabricante

A principal característica desta máquina é um campo magnético alto e a diminuição do seu volume de máquina por consequência, ocorre um grande aumento na densidade de energia em seu sistema e um alto torque por volume.

Para que o gerador forneça energia elétrica é necessária a transferência de energia mecânica a esse gerador, o que acontece por meio do movimento de rotação a seu rotor. Assim, foi adaptado um dispositivo que permite a conexão do pedal da bicicleta ergométrica com o eixo do gerador, sendo o movimento de rotação iniciado a partir das pedaladas do usuário.

À medida que esse movimento de rotação ocorre, há variação no fluxo magnético, o qual corta as bobinas dos enrolamentos da armadura (situadas no estator), por consequência origina uma força eletromotriz nas bobinas do estator.

Quando a energia mecânica está sobre efeito de um movimento rotativo, a potência produzida é dependente do conjugado C e da rotação n, a equação 12 mostra essa relação.

$$P(\text{kW}) = \frac{C(\text{Kgf.m}) * n(\text{rpm})}{974} \quad (12)$$

Onde: P (kW) = Potência em quilowatt;

C (kgf.m) = Torque em quilograma força metro;

n = Rotação.

No projeto da bicicleta, a velocidade angular é medida em rotações por minuto (rpm), sendo a cadência definida pela rotação média atingida pelas pedaladas do usuário. A equação 13 mostra o cálculo da potência relacionada à cadência e torque.

$$P = \frac{T * 2\pi * C}{60} \quad (13)$$

Onde: P(W) = Potência em Watts;

T = Torque em N.m;

C = Cadência e rotações por minuto.

O torque é calculado pela equação 8 e a rotação é medida pelo sensor de efeito Hall, que é instalado na polia (interligada ao pedal). Com estes valores

levantados é possível calcular a potência mecânica desenvolvida durante a pedalada, sendo esta potência a variável de entrada para o conjunto Fuzzy para determinar o esforço físico aplicado.

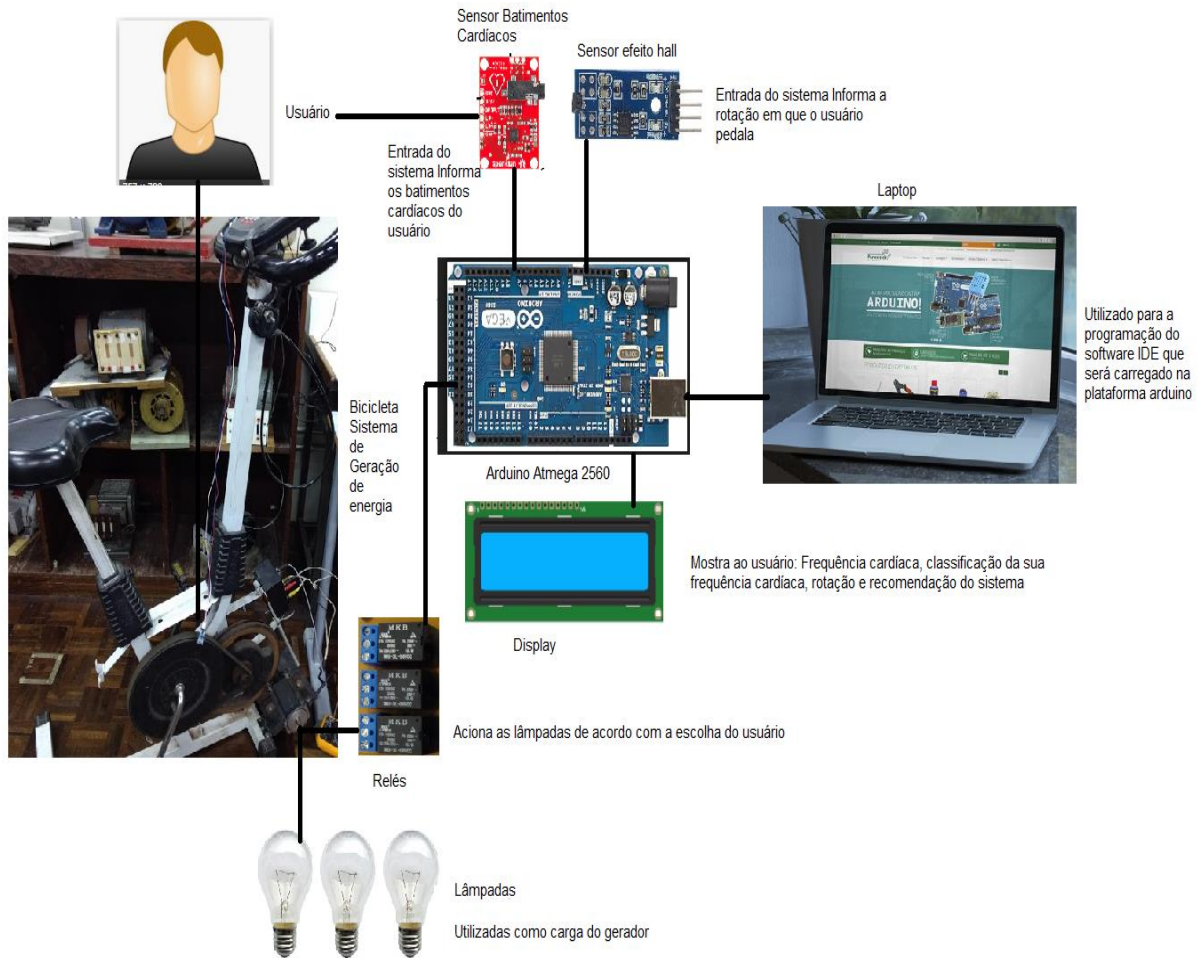
O alternador utilizado tem uma relação diretamente proporcional entre a rotação exercida a seu eixo e a tensão produzida em seus terminais de saída. A potência extraída do gerador CA é equivalente ao esforço aplicado no pedal durante o exercício físico na bicicleta ergométrica, sendo que, quanto maior o nível de potência solicitada pelo sistema, maior será o esforço físico necessário para suprir o mesmo.

A seguir é descrita a fase de construção da placa de integração.

4.3 COMPONENTES DO SISTEMA DE APOIO À DECISÃO BASEADO EM LÓGICA *FUZZY*

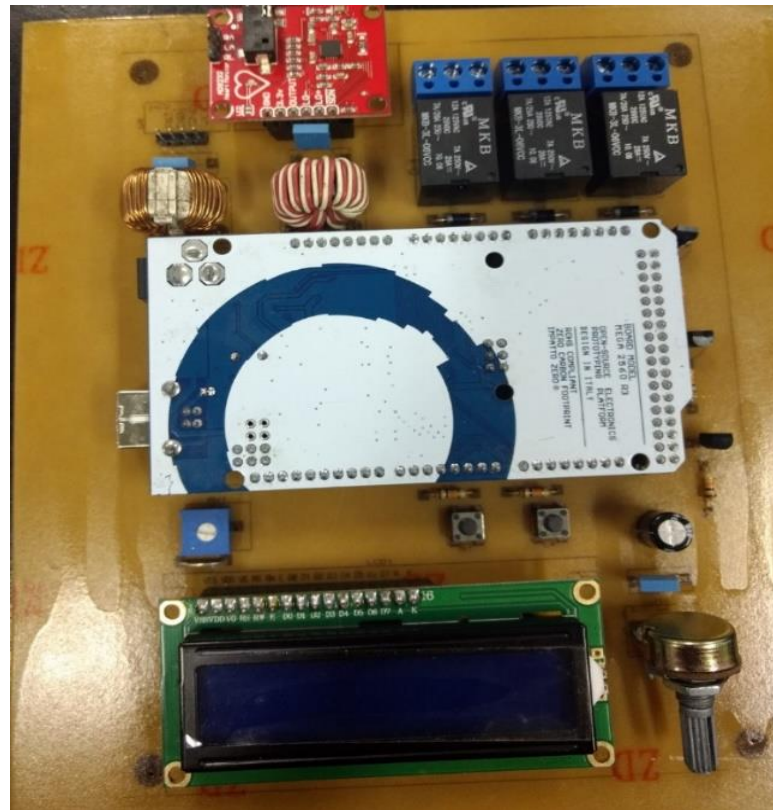
O sistema de apoio à decisão que foi construído tem como objetivo orientar o praticante de exercício físico, na bicicleta, sobre a rotação adequada que o mesmo deve pedalar. O usuário também pode escolher a intensidade da carga de acordo com que deseja e verificar sua frequência cardíaca durante o exercício. Para que funcione adequadamente foi desenvolvido um hardware que faz a integração de todos os seus componentes do sistema. A seguir cada um dos componentes da Figura 8 serão descritos.

Figura 8 - Esquemático de funcionamento do sistema.



Fonte: autoria própria.

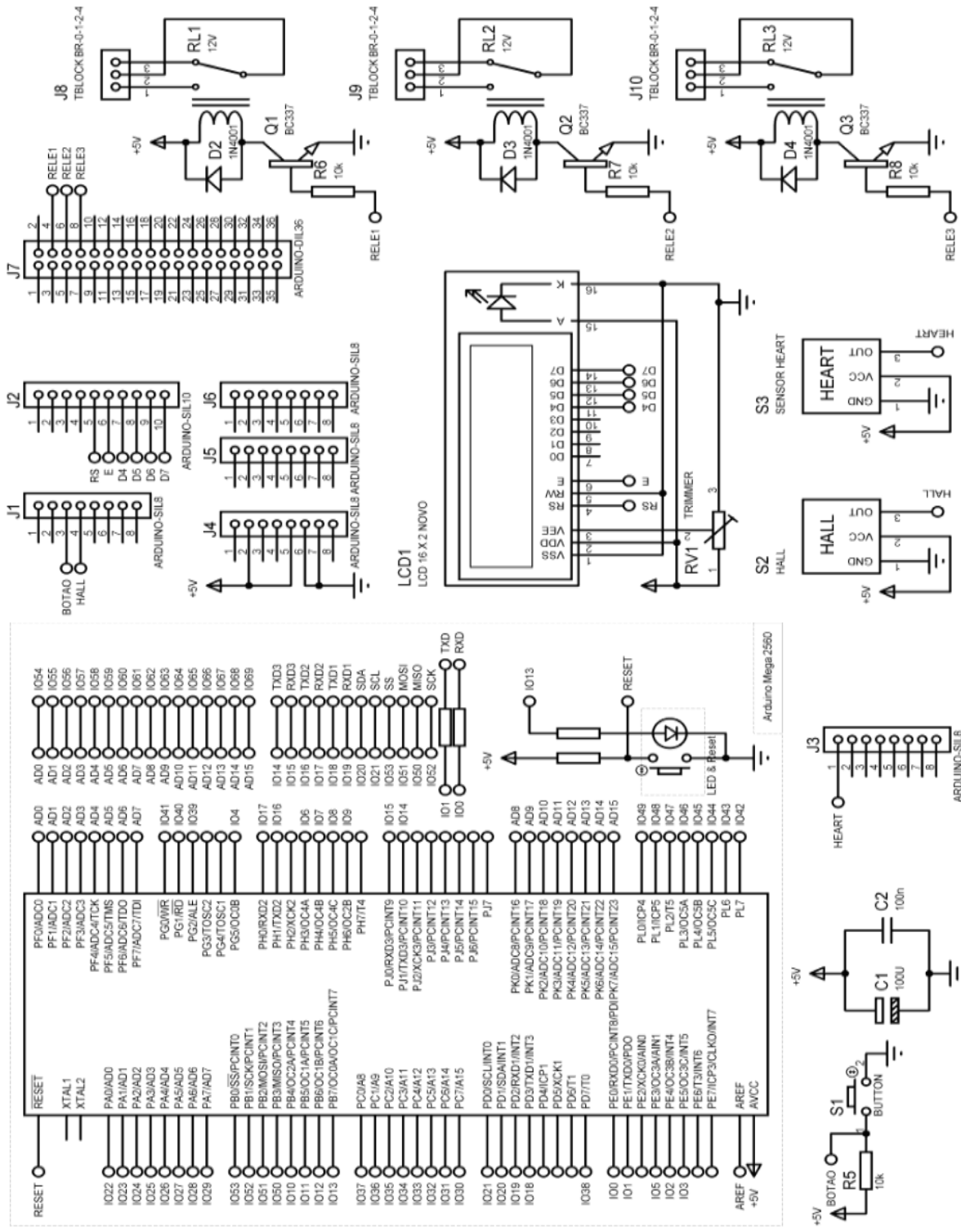
Com o objetivo de fazer a integração do ARDUINO® Mega 2560 com a interface com o usuário (botões e display), filtros, sensores e drive de controle de ligação das cargas, foi construída uma placa eletrônica conforme mostra a Figuras 9.

Figura 9 - Placa de Integração

Fonte: Autoria própria

Para a simulação do circuito, desenho do diagrama elétrico e confecção do layout da placa, foi utilizado softwares como ferramenta. O software que foi utilizado para a confecção da placa da figura 9 foi o Proteus PCB design. O diagrama elétrico da placa de integração feito nesse software é mostrado na Figura 10.

Figura 10 - Diagrama da placa de Integração



Fonte: autoria própria.

Todos os componentes do sistema serão descritos em detalhes a seguir.

4.3.1 Controlador ARDUINO® Mega 2560

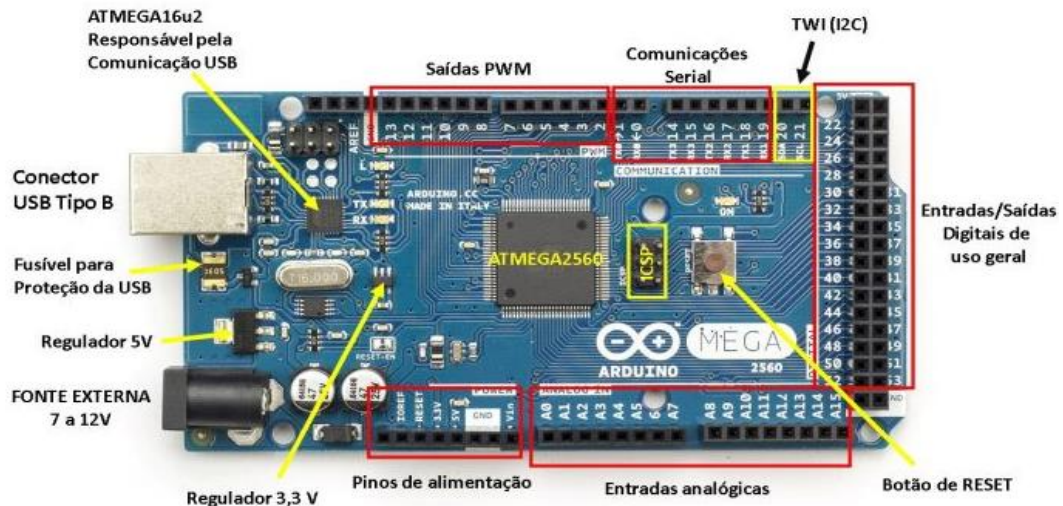
A parte superior da placa de integração é composta por 54 conectores que são pinos de barra fêmea onde será inserido o ARDUÍNO® Mega 2560 que é o controlador do sistema.

O ARDUÍNO® Mega 2560 é uma plataforma de código open-source usada para a construção de projetos eletrônicos, consistindo de uma placa de circuito físico programável um microcontrolador e um software, ou IDE (Integrated Development Environment) que é instalado no computador, utilizado para escrever e fazer upload de código feito no computador para a placa física (EMBARCADOS, 2018).

A plataforma ARDUÍNO® AT Mega 2560, possui 8 bits de arquitetura, 256 KB de Flash, 8 KB de RAM e 4 KB de EEPROM dispõe de 16 entradas analógicas, possui 54 pinos de entradas e saídas digitais onde 15 desses pinos podem ser usados como PWM. A figura 11 apresenta o detalhe deste controlador (EMBARCADOS, 2018).

A linguagem de programação no ARDUÍNO® Mega 2560 é separada em três partes principais: estrutura, valores (variáveis e constantes) e funções. A Figura 11 mostra o controlador.

Figura 11- ARDUÍNO® Mega 2560



Fonte: Embarcados Arduino-mega-2560, 2018

O sensor de efeito Hall (faz o monitoramento da rotação) e o sensor AD8232 (mede os batimentos cardíacos) são conectados na placa de integração e pelas trilhas os sinais das alterações de quaisquer desses dados, são enviados ao

ARDUINO® AT Mega 2560, um conectado ao pino de entrada digital 3 e o outro ao pino de entrada analógica A0.

Os dados que são encaminhados ao ARDUINO® AT Mega 2560, são a frequência cardíaca e a carga que está acoplada ao pedal bicicleta, essas informações são utilizadas como entradas para o software criado enviar uma saída, que é a orientação da rotação que o usuário deve pedalar.

Existem dois botões e um potenciômetro instalados na placa de integração que estão ligados nas entradas digitais e analógicas do ARDUINO® e que têm as funções de escolher o usuário, a carga na qual esse usuário irá pedalar e também de solicitar a gravação dos dados como a frequência cardíaca média, a carga, as funções de pertinência, a saída *Fuzzy* e a rotação média do treino, conforme pode ser visto na figura 13. A comunicação entre o hardware e o software IDE instalado no computador é feita via porta USB serial.

Para concluir, foi utilizado o ARDUINO® Atmega 2560 neste trabalho devido o mesmo possuir 256 KB de memória Flash, apresentando capacidade maior em relação ao Uno. Outro ponto a ser considerado, é que ARDUINO® Uno apresentou dificuldade durante o processo em *debugar* em tempo real a variável batimento cardíaco. A arquitetura do Atmega 2560 possibilitou melhor desempenho do software de monitoramento desenvolvido.

4.3.2 Display Digital e Interface com o usuário

Na placa de integração foram inseridos mais 15 pinos de barra fêmea para conectar o display. O usuário do sistema verá pelo display seu desempenho durante a pedalada como, a frequência cardíaca, a carga que escolheu para pedalar, a rotação adequada para cadência e a recomendação *Fuzzy* para aumentar, manter ou diminuir a rotação.

A apresentação das informações do usuário são divididas em três telas sendo:

- A primeira tela a ser apresentada é a de seleção do usuário. Esta tela é selecionada pelo botão de seleção do usuário, apertando o mesmo com um pulso. A figura 12 exibe a tela de seleção de usuário e a figura 13 demonstra a interface com o usuário.

Figura 12 – Tela 1 do Display para ARDUÍNO® 16X2



Fonte: autoria própria.

• A segunda tela é mostrada: a carga escolhida, a frequência cardíaca, a rotação, a recomendação Fuzzy de aumentar, manter ou diminuir a rotação e a saída *crisp*. O acesso à tela é feito pelo botão de troca de tela apertando o mesmo com um pulso. A troca da carga é feita pelo botão de troca de carga (potenciômetro), à medida que o mesmo é girado os relés de acionamento da carga são acionados. A figura 13 exibe a tela 2 e os três botões.

Figura 13 – Tela 2 do Display para ARDUÍNO® 16X2



Fonte: autoria própria.

• A terceira tela exibe os valores médios atingidos durante o exercício como: a frequência cardíaca, a carga, a rotação e a classificação da frequência cardíaca

como mínima, ideal e máxima como demonstra a figura 14. A entrada nesta tela é feita pelo botão de troca de tela apertando o mesmo com dois pulsos.

Figura 14 – Tela 3 do Display para ARDUÍNO® 16X2



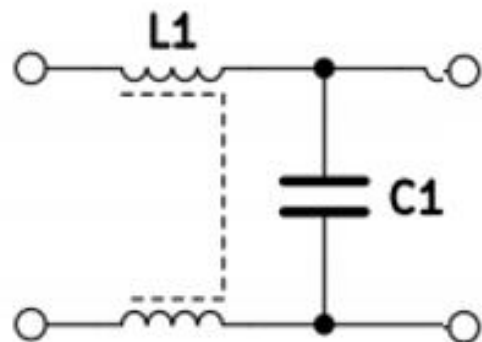
Fonte: autoria própria.

4.3.3 Filtro de Modo Comum

Devido a transmissão de energia eletromagnética que a placa de integração recebe do computador, da fonte de energia ou no chaveamento dos relés pelo circuito de alimentação ocorre a interferência eletromagnética. Interferência eletromagnética é um termo empregado aos distúrbios gerados pelos circuitos internos dos equipamentos e que podem gerar resultados indesejadas como falha no sinal de comunicação principalmente nos sensores da placa.

Para evitar esse problema foi inserido na placa de integração, em paralelo com a alimentação dos sensores, filtros de modo comum que são inseridos na linha de alimentação de energia para minimizar o fluxo de ruído. Este filtro pode ser observado na figura 15 abaixo.

Figura 15 - Filtro LC de modo comum



Fonte: autoria própria.

Os dois indutores utilizados nesta placa tem valores de 2mH e são de núcleos toroidais, onde são enrolados dois indutores em cada núcleo com o mesmo número de espiras (Figura 15). Esses indutores fornecem um caminho de alta impedância para o ruído de modo comum.

A função desse Indutor toroidal na linha de alimentação de cada sensor é de atenuar o ruído de modo comum e os fluxos magnéticos criados pelos condutores de modo diferencial (sinal e retorno) se cancelam no indutor, e assim (idealmente) ele não tem efeito no sinal desejado.

Os dois capacitores utilizados em paralelo com os indutores tem o valor de 10 μ F, são de poliéster e ambos têm a função de reduzir o ruído de modo diferencial.

4.3.4 Sensores: Sensor de Efeito Hall e Sensor de Frequência Cardíaca

Sensor é um dispositivo sensível a um fenômeno físico. Via esta sensibilidade, os sensores enviam um sinal digital podendo abrir ou fechar contatos, ou um sinal analógico para dispositivos de medição e controle (SILVEIRA; SANTOS,2001). Os sensores utilizados neste projeto são de efeito Hall e de batimentos cardíacos.

O sensor de efeito Hall instalado próximo do sistema de transmissão da bicicleta mede a rotação em que o ciclista pedala. Este sensor faz a alteração do valor de sua saída de acordo com a proximidade de um campo magnético gerado por um ímã montado na polia de transmissão conforme mostra a Figura 16. Cada vez que o ímã passa no ponto onde está preso o sensor ele comuta e assim é calculada a rotação que o ciclista pedala.

Figura 16- Sensor de efeito Hall



Fonte: autoria própria.

O sensor de batimento cardíaco (FC) AD8232 utilizado no projeto é um monitor de frequência cardíaca que funciona como um amplificador operacional para obter e filtrar a atividade elétrica do coração.

A medição dos batimentos cardíacos é feita durante a atividade de pedalar por 3 eletrodos, que ficam em contato com a pele do usuário do sistema e conectados via cabo no processador AD8232 que mede a atividade elétrica do coração. Esta atividade elétrica pode ser estruturada como um eletrocardiograma, com saída de leitura analógica que é a entrada do controlador ARDUINO® AT Mega 2560.

A Figura 17, retrata o sensor de batimento cardíaco, a alimentação VCC do sensor é igual a 3.3 V, a leitura de saída é analógica e existe um LED que pulsa ao ritmo do batimento cardíaco.

Figura 17- Sensor de medição de frequência cardíaca



Fonte: autoria própria.

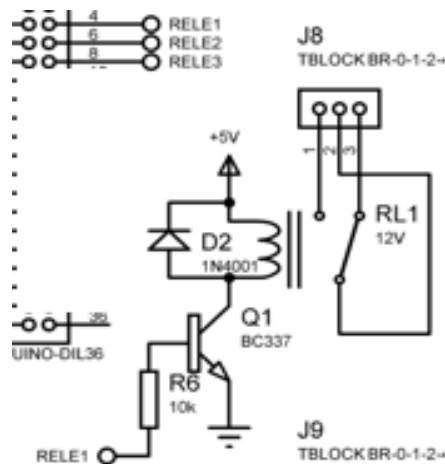
4.3.5 Drive de Acionamento das Cargas

Para inserir uma carga ao pedal da bicicleta tornando, desta maneira, o esforço aplicado pelo usuário ao pedal maior, foram ligadas três lâmpadas de 100 W na saída do gerador.

Como há a necessidade de acionar essas cargas externas foi desenvolvido uma interface para isolar os estágios de controle e potência, pois a carga exige uma corrente maior que a plataforma ARDUÍNO® pode suportar. Conforme mostra a figura 18 foi utilizado, para o acionamento da carga, um relé de bobina 6 V 30 mA com contato 250 V 10 A para carga resistiva.

O comando para ligar o relé é feito pelo transistor devido ao relé utilizar 30 mA para comutar seus contatos e para isto foi usado o transistor BC337 que atende a corrente de condução e as especificações do projeto. A base do transistor está conectada ao pino digital do ARDUÍNO® que fará o controle do chaveamento desse transistor fornecendo 5V. A Figura 18 mostra o projeto do drive de acionamento das cargas acopladas ao gerador.

Figura 18 - Drive de proteção dos pinos de saída do ARDUÍNO®



Fonte: autoria própria.

O resistor R6 tem a função de limitar a corrente no pino digital da plataforma ARDUÍNO® o dimensionamento deste resistor a equação (14):

$$R6 = \frac{(Vs - Vbe) * \beta_{min}}{Ib} \quad (14)$$

Onde: R6 - Resistência no terminal base do transistor.

Vs - Tensão fornecida no pino digital do arduino.

Vbe - Queda de tensão entre base e emissor.

Ib – Corrente de base.

O diodo D2 (Figura 18) em paralelo com a bobina tem a função de evitar que o pino do ARDUÍNO® e o transistor se danifiquem devido a liberação do eletroímã interno do relé que induz cargas sobre o circuito gerando uma corrente de pico reversa, sendo assim, haverá uma baixa resistência no diodo e alta no transistor dissipando, assim, a corrente de retorno.

4.4 SIMULAÇÃO DO SISTEMA DE APOIO À DECISÃO NO MATLAB

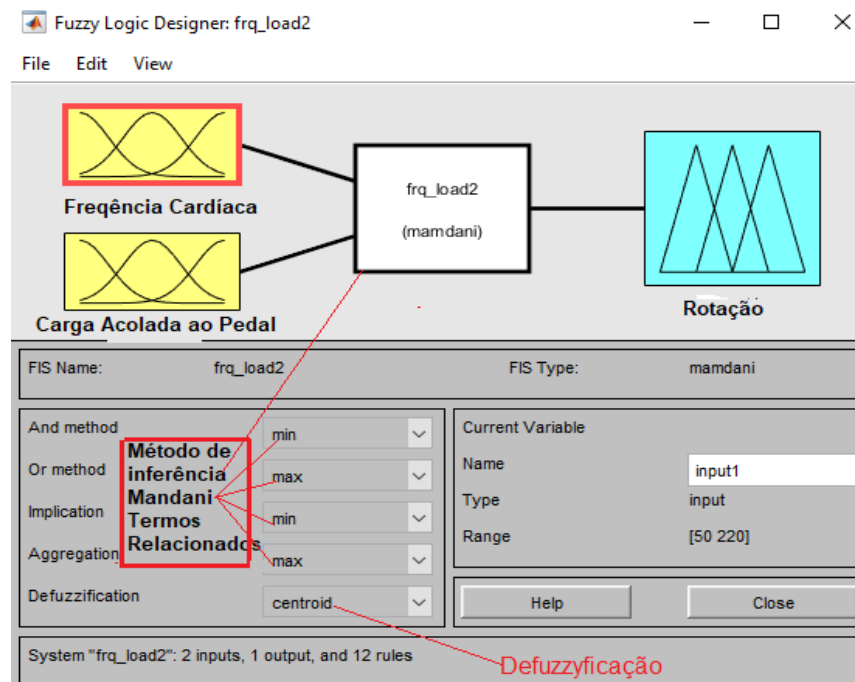
Para fins de melhor análise, refinar e validar o sistema de recomendação desenvolvido neste projeto, foi utilizado o software simulador especialista em tratamento de matrizes e sinais o MATLAB®. Nele, foi utilizado uma *Fuzzy Logical Toolbox* própria para a implementação da lógica *Fuzzy* que irá gerar diagramas da inferência para a situação desejada. Os resultados simulados no MATLAB® serviram de base para comparar com os resultados provenientes do Arduino.

A ferramenta *Fuzzy Logical Toolbox* do MATLAB®, dispõe de recursos para criar o conjunto de regras, definir as funções de associação e analisar o comportamento de um sistema de inferência *Fuzzy*.

O *Fuzzy Logic* toolbox do MATLAB® dispõe de dois métodos de inferência: Método de MAMDANI e o Método de Sugeno e para defuzzificação oferece vários métodos (AMENDOLA; SOUZA; BARROS, 2005).

Esse trabalho utilizou o Método Mandani para inferência e centro de gravidade para a defuzzificação, devido a simplicidade e eficiência desses métodos. A Figura 17 mostra a configuração básica da metodologia adotada.

Figura 19 - Configuração básica do uso da ferramenta *Fuzzy Logical Toolbox* com as especificações estabelecidas



Fonte: autoria própria.

4.5 IMPLANTAÇÃO DA LÓGICA FUZZY NA PLATAFORMA ARDUINO

Para o software que usa a lógica *Fuzzy* em seu controle foi utilizada a biblioteca e-FLL² para Arduino escrita em C++/C, que usa apenas a biblioteca padrão da linguagem C "*stdlib.h*", esta biblioteca foi criada pela *Robotic Research Group* (RRG) na Universidade Estadual do Piauí. A biblioteca utiliza o processo (MAX-MIN) e (Mínimo de MAMDANI) para a inferência e composição e (Centro de Área) para a defuzzificação, no universo contínuo (ZEROKOL, 2012).

Com a finalidade de fazer um sistema que seja capaz de monitorar e orientar a rotação em que o usuário deve pedalar em uma bicicleta ergométrica foram construídas regras *Fuzzy* que classificam a frequência cardíaca e a carga desse usuário. A frequência cardíaca é classificada como: mínima, ideal e máxima e a carga é identificada como: leve, média e pesada.

Para montar os conjuntos *Fuzzy* batimento cardíaco: mínimo, ideal e máximo no código deste trabalho e na simulação pelo MATLAB®, foi utilizada a equação número (8), onde a idade determina a frequência cardíaca máxima (FCM) do usuário. A equação (9) estabelece o ponto em que se tem 60% da FCM, formando os valores do conjunto *Fuzzy* mínimo. Na equação (10) tem-se 70% da FCM, estes valores formam o conjunto *Fuzzy* ideal. A equação (11) determina o ponto de 85% da FCM, estes valores formam o conjunto *Fuzzy* máximo. Devido a idade ser uma variável que forma os conjuntos *Fuzzy* batimento cardíaco, ela está fixada no código e cada vez que muda a idade do usuário é necessário efetuar a mudança dos conjuntos *Fuzzy* no código do sistema.

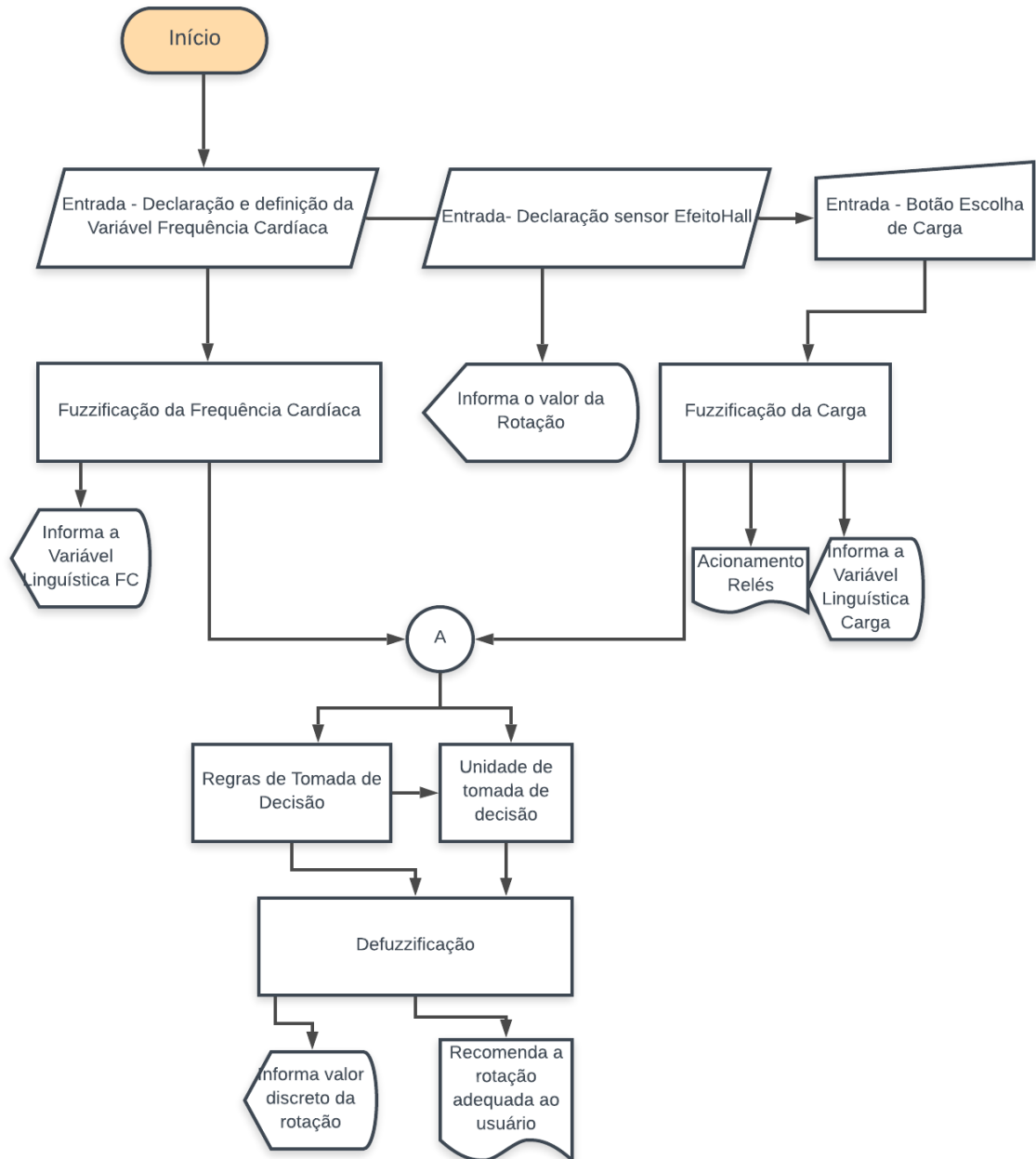
Após a utilização de todos os dados processados pelo software instalado no ARDUÍNO®, as informações dos sinais dos sensores são enviadas para a placa de controle, que informa a frequência cardíaca e classifica a mesma pelo processo de fuzzificação, mostra a carga e orienta qual a rotação o utilizador do sistema deve pedalar.

4.5.1 Estrutura do Programa de Apoio a Decisão Baseado em Lógica Fuzzy

² Fonte: <https://github.com/zerokol/eFLL>

A figura 20 representa um fluxograma que ilustra o funcionamento do programa.

Figura 20 - Fluxograma do software de Apoio a Decisão



Fonte: autoria própria.

Conforme apresentou o fluxograma do software de apoio à decisão, a construção do programa realiza todas as etapas de fuzzificação, inferência e

defuzzificação das variáveis do sistema, necessárias ao bom funcionamento do mesmo.

Essa dissertação utilizou o software MATLAB[®] para todo o desenvolvimento do sistema Fuzzy. Antes de elaborar o programa para o Arduino, o sistema Fuzzy foi testado e validado no MATLAB[®], de onde provém as figuras e os resultados aqui mostrados.

Após os testes e resultados no MATLAB[®] conforme visto anteriormente, o algoritmo de apoio à decisão baseado na lógica *Fuzzy* foi desenvolvido no software ARDUÍNO[®] IDE (*Integrated Development Environment*), o qual este é instalado em um laptop, sendo executado em ambiente Windows. A linguagem de programação usada pelo ARDUÍNO[®] IDE é o C++ e fundamentado em *open source* software.

Com a finalidade de implementar a lógica *Fuzzy* no software IDE e no algoritmo do projeto, achou-se a necessidade de procurar por uma biblioteca *Fuzzy* e durante as pesquisas foi encontrada uma biblioteca *Fuzzy* denominada eFLL como citado anteriormente no item 4.5 para a plataforma Arduino e também para qualquer Sistema Embarcado. A biblioteca eFLL, não tem restrições de quantidade de conjuntos *Fuzzy*, regras *Fuzzy* e entradas ou saídas (ZEROKOL, 2012).

De acordo com o Software livre brasil, 2014 a biblioteca eFLL é dividida em objetos que são usados no processo, os quais são:

- Objeto *Fuzzy*: integra todo o Sistema *Fuzzy*, através dele é permitido operar os Conjuntos *Fuzzy*, as Regras Linguísticas, entradas e saídas.

- Objeto *FuzzyInput*: associa todos os Conjuntos *Fuzzy* de entradas que pertencem ao mesmo domínio.

- Objeto *FuzzyOutput*: associa todos os Conjuntos *Fuzzy* de saída que pertencem ao mesmo domínio.

- Objeto *FuzzySet*: é um dos mais importantes objetos da biblioteca *Fuzzy*, com ele é possível formar cada conjunto do sistema. A biblioteca comporta as funções de pertinência triangular, trapezoidal e singleton, que são criadas com base nos pontos a, b, c e d que são passados por parâmetro no construtor *FuzzySet* (*float a, float b, float c, float d*).

- Objeto *FuzzyRule*: é utilizado para criar a base de regras do objeto *Fuzzy*, Instanciado com *FuzzyRule* `fr = new FuzzyRule(ID, antecedent, consequent)`.

- Objeto `FuzzyRuleAntecedent`: é utilizado para formar o objeto `FuzzyRule`, incumbido por montar a expressão condicional do antecedente de um `FuzzyRule`, exemplo:

```
FuzzyRuleAntecedent* iffrqcarmin20ANDleve = new FuzzyRuleAntecedent();
```

- Objeto `FuzzyRuleConsequent`: é utilizado para formar o objeto `FuzzyRule`, incumbido por montar a expressão de saída de um `FuzzyRule`, exemplo:
`FuzzyRuleConsequent* thenSisACM = new FuzzyRuleConsequent();`

- `bool fuzzify()`: é usado para ativar o processo de Fuzzyficação, composição e inferência.

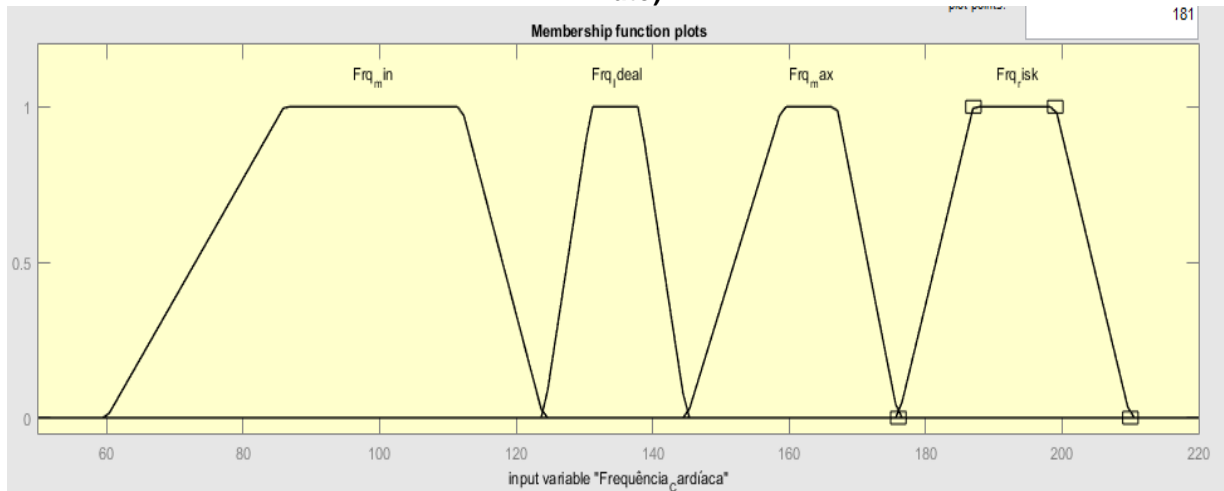
- `float defuzzify(int ID)`: É usado para concluir o processo de Fuzzyficação, chama a variável de saída defuzzificada.

4.5.2 Entradas e Saída do Sistema

Como entradas do sistema *Fuzzy* foram escolhidas duas variáveis. A primeira variável, que por pesquisa é a variável mais importante, é o batimento cardíaco pois, estima o esforço que representa uma atividade de trabalho para um indivíduo que está pedalando. A frequência foi classificada como mínima, ideal, máxima e de risco cardíaco para o exercício físico baseadas na metodologia de Karnoven como, visto no capítulo anterior.

O universo de discurso foi determinado entre 50 e 220 batimentos cardíacos por minuto conforme mostra a figura 19 e as funções de pertinência para a variável mencionada acima são: `(Frq_min)`, `(Frq_ideal)`, `(Frq_max)` e `(Frq_risk)`. A classificação dessas variáveis serão mostradas em um display.

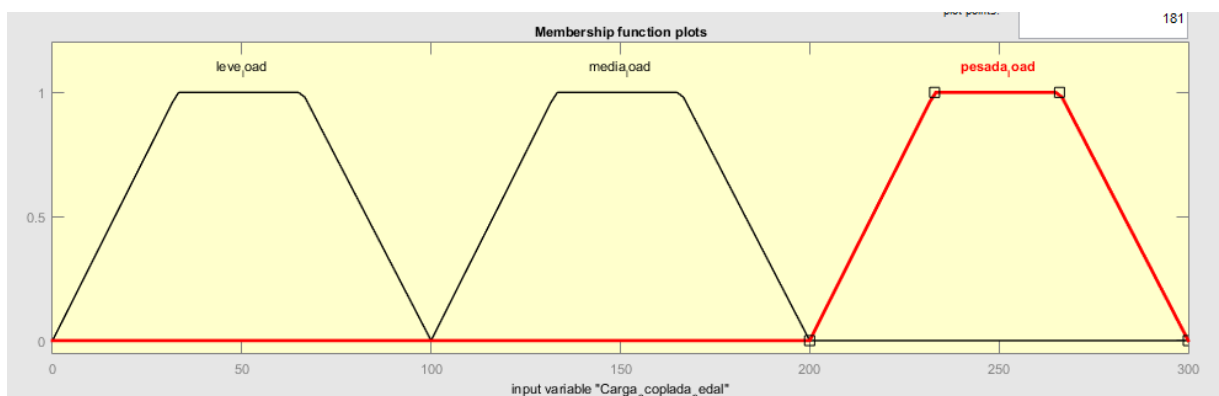
Figura 21 - Variável de entrada batimentos cardíacos por minuto (entre 50 e 220 batimentos por minuto)



Fonte: autoria própria.

A segunda variável é a carga, importante para intensificar o trabalho muscular do utilizador do sistema e o universo de discurso desta variável está entre 0 e 300 watts que são cargas acopladas ao gerador instalado no sistema. Essas cargas foram classificadas em carga leve (CL), carga média (CM) e carga pesada (CP). A Figura 22 abaixo mostra a função de pertinência da variável de entrada carga.

Figura 22 - Variável de entrada carga acoplada ao gerador universo de discurso entre 0 e 300 watts.



Fonte: autoria própria.

Para criar as variáveis de entrada vistas acima no software IDE do ARDUÍNO® utilizando a biblioteca eFLL, na função Fuzzy setup foi usado o objeto FuzzyInput e seus endereços (1) e (2) devem ser dados como parâmetros do construtor FuzzyInput de acordo com as linhas de códigos abaixo:

```
FuzzyInput* BPM_Fuzzy = new FuzzyInput(1);
```

```
FuzzyInput* load = new FuzzyInput(3);
```

A construção dos conjuntos *Fuzzy* à partir das variáveis de entrada é feita utilizando o Objeto *FuzzySet*, cuja função é moldar as variáveis de entrada frequência cardíaca e carga para: frequência cardíaca, mínima, ideal, máxima e de risco e carga leve, média e pesada. Este trecho do código do projeto demonstra a modelagem das variáveis de entrada:

```
// Frequencia cardiaca (20 a 25 anos)
FuzzySet* frqcar_min = new FuzzySet (60, 86, 112, 124);
FuzzySet* frqcar_ideal = new FuzzySet (124, 131, 138, 145);
FuzzySet* frqcar_max = new FuzzySet (145, 159, 167, 176);
FuzzySet* frqcar_risk = new FuzzySet (176, 187, 199, 210);
// Carga
FuzzySet* CL = new FuzzySet (0,33, 66, 100 ); // carga leve
FuzzySet* CM = new FuzzySet (100 133 166 200); // carga ideal
FuzzySet* CP = new FuzzySet (200, 233, 266, 300); // carga pesada
```

Depois de moldar os conjuntos é feita a associação da categoria dos conjuntos para a variável *FuzzyInput* *BPM* e *load_choice* como mostra o código a seguir.

```
BPM->addFuzzySet(frqcar_min);
BPM->addFuzzySet(frqcar_ideal);
BPM->addFuzzySet(frqcar_max);
BPM->addFuzzySet(frqcar_risk);
```

O valor da variável *BPM* é dada pela entrada *A0* do *ARDUINO* que está conectado do sensor de batimentos cardíacos.

```
load_choice ->addFuzzySet(leve);
load_choice ->addFuzzySet(media);
load_choice >addFuzzySet(pesada);
```

O valor da variável *load_choice* (carga) é obtida, quando o usuário escolhe pelo potenciômetro instalado na placa do Sistema, a intensidade da carga com que ele deseja pedalar. Quando a carga é selecionada, os relés correspondentes a ela são acionados acendendo as lâmpadas.

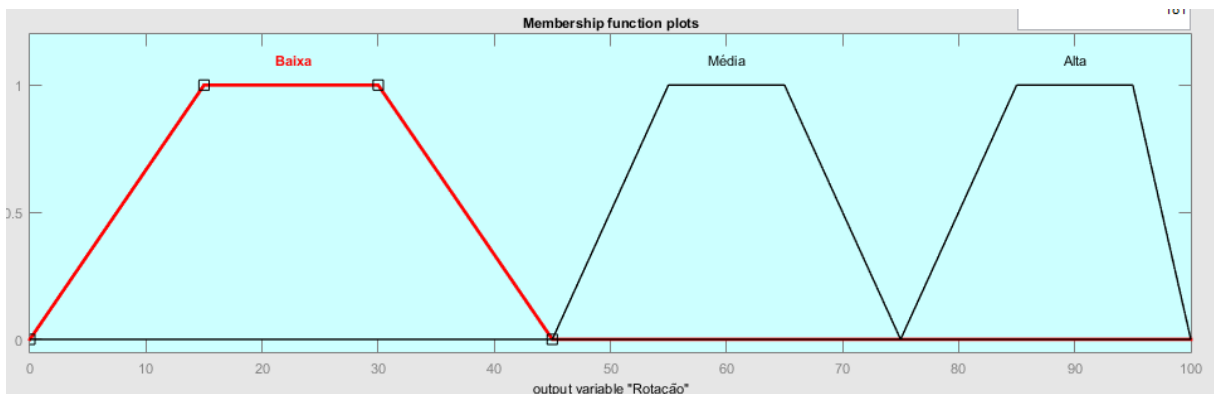
Para estipular o valor mais aproximado da carga é utilizada a equação 12 porque a potência que será entregue a carga irá variar de acordo com a rotação que o usuário pedala.

Segue abaixo a parte do Código correspondente a carga.

```
int load_choice= ( (100) * (FREQ_HALL / 100));
int analogValue = analogRead(LOAD_CHOICE_ANALOG_PIN);
load_choice = ( (100) * (FREQ_HALL / 100)) + 100* (floor(analogValue/341.3))
*(FREQ_HALL / 100);
E por fim, insere a FuzzyInput no objeto Fuzzy.
Fuzzy->addFuzzyInput(BPM);
Fuzzy->addFuzzyInput(load_choice);
```

Como saída do sistema, foi utilizada a variável rotação que orienta o usuário do sistema qual a melhor rotação que este usuário deve atingir para ter mais proveito em seu treino. Aparecerá em um display uma orientação que é resultado da saída Fuzzy para diminuir muito, diminuir, manter, aumentar e aumentar muito a rotação. Essa saída é dependente das entradas: frequência cardíaca e a da carga acoplada ao pedal. As variáveis linguísticas dadas para a rotação são mostradas na Figura 23 abaixo.

Figura 23 - Variável de saída rotação. Universo de discurso entre 0 a 100 rotações por minuto.



Fonte: Autora

Para criar as variáveis de saída vistas no software IDE do arduino utilizando

a biblioteca eFLL, utiliza-se os mesmos passos das variáveis de entrada visto anteriormente somente troca-se o objeto FuzzyInput pelo FuzzyOutput segue o exemplo abaixo com a linha de código para essa etapa.

```
FuzzyOutput* S = new FuzzyOutput(1);
FuzzySet* baixa = new FuzzySet( 0, 15, 30, 45);
S->addFuzzySet(baixa);
FuzzySet* media = new FuzzySet(45, 55, 65, 75);
S->addFuzzySet(media);
FuzzySet* alta = new FuzzySet(75 85 95 100);
S->addFuzzySet(alta);
Fuzzy->addFuzzyOutput(S);
```

4.5.3 Regras *Fuzzy*

Para elaboração das regras *Fuzzy* foi efetuada revisão da literatura acerca da relação entre atividade física e a frequência cardíaca. A tabela de regras *Fuzzy* é a base do conhecimento, é o banco de inteligência do sistema elaborado pelo especialista, por fim é um arranjo traduzido pelas combinações de variáveis de entrada e saída, em termos de seus conjuntos linguísticos.

Para elaborar a base de regras foram estudados artigos na área de educação física e fisioterapia e, de acordo com a frequência cardíaca e com a carga que o usuário escolheu, o sistema de apoio à decisão solicita para esse usuário aumentar, manter ou diminuir a rotação. O conjunto de saída rotação: A (rotação alta), Ma (rotação média) e D (rotação baixa), deve ser inserido no cruzamento dos conjuntos Fuzzy de entrada Frq_min, Frq_ideal, (Frq_max), (Frq_risk), CL, CM e CP. A Tabela 2 apresenta as regras.

Tabela 2 - Mapa de Regras *Fuzzy*

Carga Acionada de saída	Frequência Cardíaca (FC)			
	Frq_min	Frq_ideal	Frq_max	Frq_risk
CL	A	A	Ma	D
CM	A	Ma	D	D
CP	Ma	D	D	D

Fonte: autoria própria.

A Tabela 2 representou o conjunto de regras para a estratégia de apoio a decisão, a primeira etapa foi a identificação das variáveis de entrada com a associação referente as variáveis linguísticas, que são: Frq_min, Frq_ideal, Frq_max e Frq_risk representam os conjuntos *Fuzzy* dos valores das variáveis de entrada frequência cardíaca mínima, frequência cardíaca ideal, frequência cardíaca máxima e frequência cardíaca de risco e também CL, CM e CP que representam os conjuntos *Fuzzy* dos valores das variáveis de entrada carga leve, carga média e carga pesada.

Realizada a identificação das variáveis de entrada que corresponde ao módulo de fuzzificação o método de inferência avalia as regras da seguinte forma:

1. Se a frequência cardíaca é mínima (Frq_min) e carga é leve (CL) Então rotação é alta (A);
2. Se a frequência cardíaca é mínima (Frq_min) e carga é média então rotação é alta (A);
3. Se a frequência cardíaca é mínima (Frq_min) e carga é pesada (CP) então rotação é média (Ma);
4. Se a frequência cardíaca é ideal (Frq_Ideal) e carga é leve então rotação é alta (A);
5. Se a frequência cardíaca é ideal (Frq_Ideal) e carga é média (CM) então rotação é média (Ma);
6. Se a frequência cardíaca é ideal (Frq_Ideal) e carga é pesada (CP) então rotação é baixa (D);
7. Se a frequência cardíaca é máxima (Frq_max) e carga é leve (CL) então rotação é média (Ma);
8. Se a frequência cardíaca é máxima (Frq_max) e carga é média (CM) então rotação é baixa (D);

9. Se a frequência cardíaca é máxima (Frq_max) e carga é pesada (CP) então rotação é baixa (D);

10. Se a frequência cardíaca é risco (Frq_risk) e carga é leve (CL) então rotação é baixa (D);

11. Se a frequência cardíaca é risco (Frq_risk) e carga é média (CM) então rotação é baixa (D);

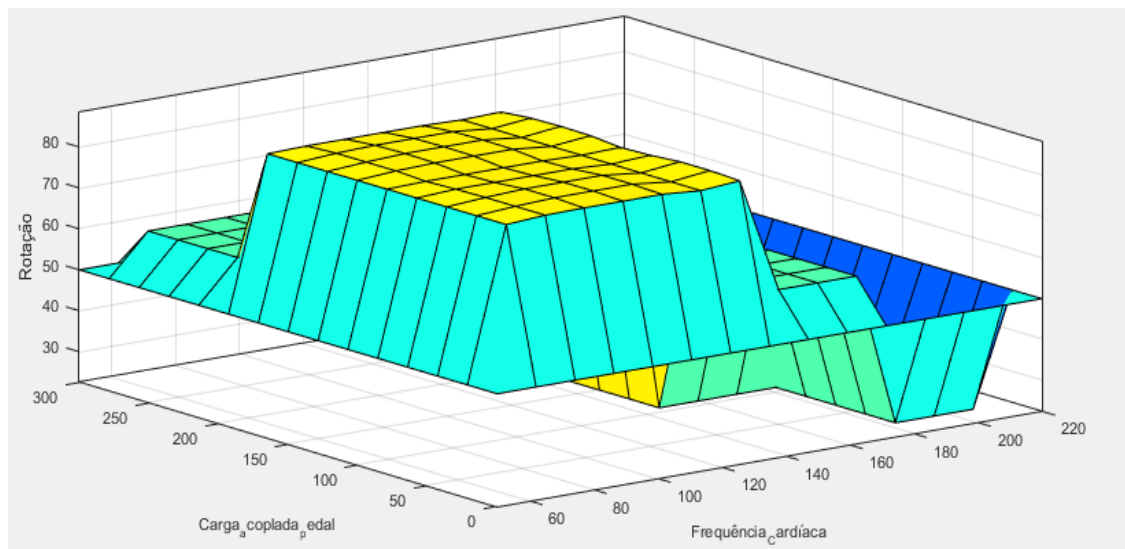
12. Se a frequência cardíaca é risco (Frq_risk) e carga é pesada (CP) então rotação é baixa (D).

As doze regras acima possuem dois antecedentes como visto, a frequência cardíaca associada com a carga e o resultado é a combinação desses antecedentes. A parte consequente à rotação, foi formada por um conjunto de ações que foram geradas com o disparo dessas regras.

No cruzamento entre a frequência cardíaca e a carga acionada está um conjunto de saídas que são sugestão de qual a rotação é adequada para o usuário do sistema ter um bom aproveitamento do seu treino e essa rotação será mostrada em um display.

Fundamentado na base de regras, é gerada uma superfície, conforme apresentada na Figura 24.

Figura 24– Base de Regras no MATLAB®



Fonte: autoria própria.

A Figura 24 mostrou a superfície 3D da variação da rotação, como função da frequência cardíaca e da carga e foi gerada a partir da base de regras determinadas na Tabela 2. Assim sendo, o conjunto de valores da frequência cardíaca e da carga, tem-se como resultado a inferência de um valor no intervalo [0,1] que representa a rotação adequada que o usuário deve pedalar. Nesta lógica, é possível obter uma saída do sistema de inferência, por exemplo, como mostra a superfície, uma frequência cardíaca de 90 bpm e carga de 90 W, e, após a defuzzificação encontra o valor de 80 rpm que é denominada uma rotação alta.

Observou-se também que o platô da superfície está no conjunto das frequências cardíacas mínima e na carga leve, estando assim de acordo com as regras que à medida que a frequência cardíaca é mínima ou ideal a rotação deverá aumentar e à medida que a frequência cardíaca aumenta a rotação deverá diminuir. O comportamento do sistema, que é o que a superfície mostra, está de acordo com o esperado. Esta é a interpretação que permite o suporte a decisão.

Após a demonstração da figura 24 feita no MATLAB® necessita-se inserir as bases de regras no IDE do ARDUÍNO® e utilizar biblioteca eFLL. O trecho do código abaixo mostra como é feita a montagem das regras no IDE do ARDUÍNO®.

Cada regra é gerada por uma condição antecedente responsável por montar a expressão condicional (SOFTWARE LIVRE BRASIL, 2014).

```
FuzzyRuleAntecedent*iffrqcarmin20ANDleve = new
FuzzyRuleAntecedent();
```

O recurso `joinWithAND` é utilizado para montar a expressão E (and) e também faz composições lógicas entre os conjuntos Fuzzy.

```
iffrqcarmin20ANDleve->joinWithAND(frqcarmin20, leve);
```

O próximo passo é construir o objeto responsável pela saída.

```
FuzzyRuleConsequent* thenSisA = new FuzzyRuleConsequent();
thenSisA ->addOutput(A);
```

O último passo é incluir a regra na base de regras. Depois de criar um objeto *FuzzyRule*, utilizou a ferramenta `addFuzzyRule(FuzzyRule* FuzzyRule)`

```
FuzzyRule* FuzzyRule1 = new FuzzyRule(1, iffrqcarmin20ANDleve,
thenSisA);
Fuzzy->addFuzzyRule(FuzzyRule1);
```

Observação: Este mesmo processo de programação se repete para as 12 regras do projeto.

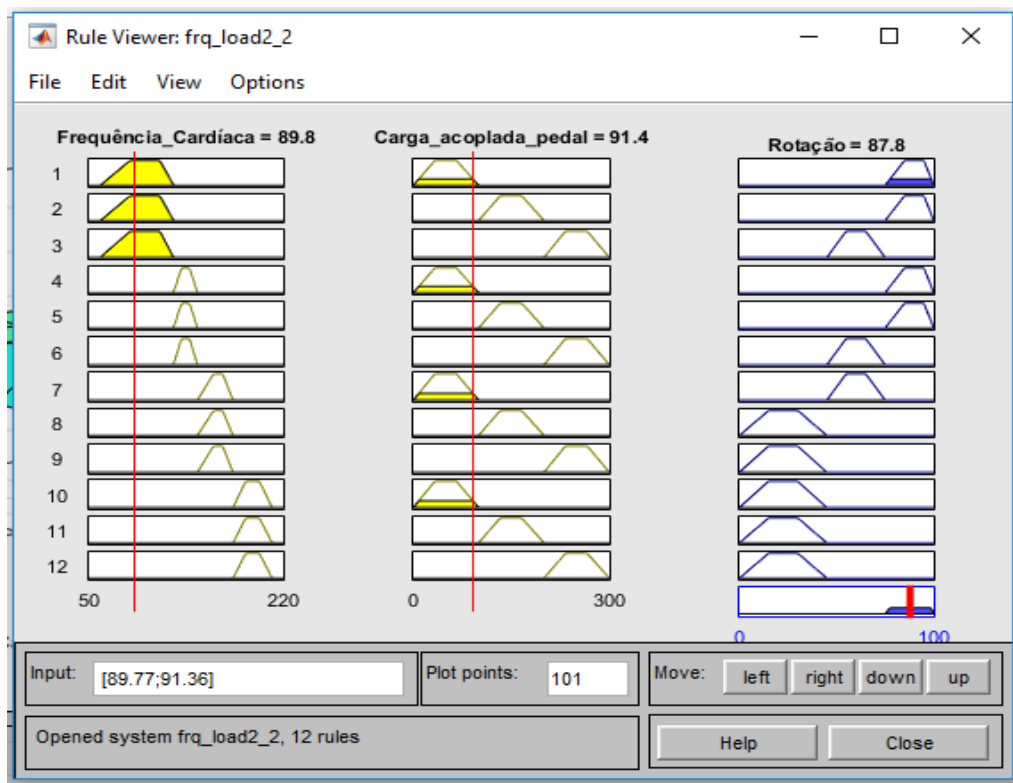
A defuzzificação utilizada nessa dissertação, tanto na simulação com o MATLAB® quanto com a implementação do arduino utilizando a biblioteca IDE, é o método MAMDANI centróide como citado anteriormente. O valor do consequente é calculado como o centro de área da figura resultante da união dos conjuntos Fuzzy no domínio da saída (ENGEL, 2019).

Os valores de $\mu(x)$ indicam o grau de compatibilidade e são multiplicados pelo valor x e é feita uma somatória deste valor obtido em cada uma das regras e o total é dividido pela somatória dos graus de pertinência, valor resultante com o conceito modelado pelo conjunto Fuzzy X_0 conforme mostra a equação 15.

$$x_0 = \frac{\sum x \cdot \mu(x)}{\sum \mu(x)} \quad (15)$$

A Figura 25 mostra a defuzzificação pelo método Centro de Área.

Figura 25 - Defuzzificação utilizando o método do centro de área.



Fonte: autoria própria.

Na Figura 25, construída no ambiente computacional do MATLAB®, pode ser observada a unificação das saídas dos conjuntos consequência de todas as regras Fuzzy aplicadas no trabalho.

4.6 COLETA DOS DADOS

Para coleta das variáveis de entrada, frequência cardíaca e carga, se fez necessária a aplicação de testes de esforço físico na bicicleta ergométrica. Para tanto, foram escolhidos 3 usuários para compor os eventos, sendo os perfis destes apresentado no Quadro 4.

Quadro 4 - Perfil dos Usuários

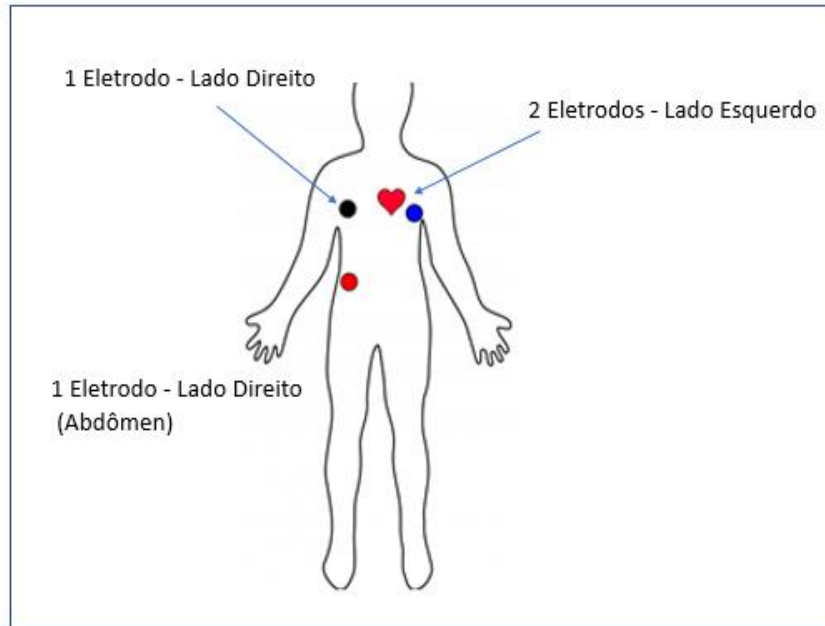
Usuário	Idade	Sexo	Situação – Atividade Física
U1	23	Masculino	Não sedentário
U2	37	Feminino	Não sedentário
U3	40	Masculino	Não sedentário

Fonte: Autora

Nos preparativos para o teste na bicicleta ergométrica foram repassadas as seguintes orientações aos usuários:

- (1) Colar os eletrodos no corpo conforme os pontos indicados na Figura 26.

Figura 26– Pontos para Conexão dos Eletrodos no Corpo Humano



Fonte: autoria própria.

(2) Após conexão dos eletrodos, sentar na bicicleta ergométrica de forma que o cabo de conexão entre eletrodo e o sensor de batimento cardíaco não encostem nas pernas do usuário, para que não haja interferência nos resultados do teste;

(3) Orientar o usuário em relação ao uso do display digital no momento de escolha da carga e da confirmação do usuário. Isto deve ser feito por meio de aperto do botão, que está instalado na placa de circuito impressora integradora (Observação: não deve ser feita a escolha com aperto na tela do display);

(4) Orientar o usuário em relação ao monitoramento da frequência cardíaca e da classificação desta (mínima, ideal e máxima) disponíveis na interface na tela do display. Também há necessidade da observância, por parte do usuário e da pesquisadora, em relação a variável de saída de rotação adequada de acordo com as entradas (frequência e carga escolhida), também disponível no display.

A partir dos testes foram coletados os dados das variáveis de entrada: frequência cardíaca, carga média utilizada pelo usuário e as variáveis de saída recomendadas pelo sistema desenvolvido: rotação. Estes dados foram organizados e tabulados para análise comparativa com os dados obtidos na simulação com o software MATLAB®.

5 RESULTADOS E DISCUSSÕES

5.1 RESULTADOS DOS TESTES NO LABORATÓRIO

Para obter os resultados do sistema desenvolvido, foi construído um código que calcula os valores médios das variáveis de entrada e de saída, e, posteriormente são mostrados no display digital. Com estes resultados é efetuada a simulação no ambiente computacional MATLAB® com intuito de comparar as funções de pertinência: de entrada, de saída e a defuzzificação que é a rotação adequada para pedalar.

Com o objetivo de verificar a eficácia do sistema de recomendação construído, foram gerados gráficos no MATLAB® com as funções de pertinência de entrada e de saída, bem como o valor numérico da defuzzificação. Posteriormente, os valores obtidos dessa simulação foram confrontados com os dados obtidos nos testes de esforços físicos dos 3 eventos, sendo estes dados obtidos pelo display.

Para a coleta de dados realizou-se testes com 3 usuários na bicicleta ergométrica. Foram medidos: batimento cardíaco, carga, rotação e tensão elétrica durante os testes. Ressalta-se que a tensão foi medida por meio de instrumentos de medição (multímetro), sendo que esta variável não está contemplada na plataforma ARDUINO®.

Individualmente foram coletados os dados particulares dos usuários durante o teste da atividade física na bicicleta, sendo esses dados mostrados pelo display instalado na placa de integração que está conectado por um suporte na bicicleta ergométrica utilizada no estudo.

Posteriormente efetuou-se a tabulação dos valores obtidos nos testes, assim, foi executada a comparação entre os valores simulados para validar a funcionalidade do sistema de recomendação desenvolvido.

Os valores obtidos nos experimentos utilizando sistema de recomendação na plataforma ARDUINO® para cada usuário estão tabulados na Tabelas 3, 4,5,6,7 e 8 mostradas a seguir.

Tabela 3 - Valores Obtidos nos Testes Usuário 1

Usuário	Idade	Batimento	Carga(W)	Rotação	Tensão
		cardíaco (bpm)		(rpm)	(V)
U1	23	138	90	90	105
U1	23	179	120	60	70
U1	23	170	200	50	59

Fonte: autoria própria.

Tabela 4- Valores Obtidos nos Testes Usuário 2

Usuário	Idade	Batimento	Carga(W)	Rotação	Tensão
		cardíaco (bpm)		(rpm)	(V)
U2	37	129	100	95	111
U2	37	105	140	70	82
U2	37	128	200	50	59

Fonte: autoria própria.

Tabela 5- Valores Obtidos nos Testes Usuário 3

Usuário	Idade	Batimento	Carga(W)	Rotação	Tensão
		cardíaco (bpm)		(rpm)	(V)
U3	40	152	80	80	94
U3	40	149	100	70	82
U3	40	134	150	55	64

Fonte: autoria própria.

As Tabelas 3, 4 e 5 mostram os valores médios: da frequência cardíaca, da carga e da rotação (mostrados no display durante os testes na bicicleta) para cada usuário.

Já as Tabelas 6, 7 e 8 apresentam: a idade, a classificação da frequência cardíaca, a carga, a rotação atual do usuário, o resultado numérico (defuzzificação) e a recomendação para aumentar, manter ou diminuir a rotação. Todas estas variáveis são mostradas no display.

Tabela 6 - Valores de saída do Sistema de Recomendação Usuário 1

Variáveis de Entrada				Rotação Atual	Variáveis de Saída	
Usuário	Idade	Frequência Cardíaca	Carga (W)	Rotação Atual do Usuário	Rotação Defuzificação	Orientação de Rotação
1	23	Ideal	90	90	85	Diminuir
1	23	Risco	120	60	20	Diminuir
1	23	Máxima	200	50	23	Diminuir

Fonte: autoria própria.

Tabela 7 - Valores de saída do Sistema de Recomendação Usuário 2

Variáveis de Entrada				Rotação Atual	Variáveis de Saída	
Usuário	Idade	Frequência Cardíaca	Carga (W)	Rotação Atual do Usuário	Rotação Defuzificação	Orientação de Rotação
2	37	Ideal	100	95	87	Diminuir
2	37	Ideal	140	70	85	Aumentar
2	37	Ideal	200	50	60	Aumentar

Fonte: autoria própria.

Tabela 8 - Valores de saída do Sistema de Recomendação Usuário 3

Variáveis de Entrada				Rotação Atual	Variáveis de Saída	
Usuário	Idade	Frequência Cardíaca	Carga (W)	Rotação Atual do Usuário	Rotação Defuzificação	Orientação de Rotação
3	40	Máxima	80	80	60	Diminuir
3	40	Máxima	100	70	55	Diminuir
3	40	Ideal	150	50	50	Manter

Fonte: autoria própria.

As Tabelas 6, 7 e 8 mostraram as variáveis linguísticas de entrada e também os resultados da saída do sistema de recomendação com a orientação da rotação adequada. A saída foi obtida com o cálculo do centro de gravidade das amostras depois da combinação de todas as regras conforme apresentou a equação 15.

Para que o sistema o sistema faça a orientação de aumentar, manter ou diminuir a rotação utilizou-se o código descrito abaixo:

```
if ((output - FREQ_HALL) > 0)
LCD.print("AUMENTAR");
else if ((output - FREQ_HALL) < 0)
LCD.print("DIMINUIR");
else if ((output - FREQ_HALL) == 0)
LCD.print("MANTER");
```

Onde:

Output: é a saída do sistema (rotação).

FREQ_HALL: Rotação atual atingida pelo usuário.

Tomando como exemplo a tabela 6, linha1 a saída do sistema recomendou uma rotação de 85 rpm, como o usuário 1 estava pedalando a uma rotação de 90 rpm que é acima do recomendado o sistema orientou para diminuir a rotação.

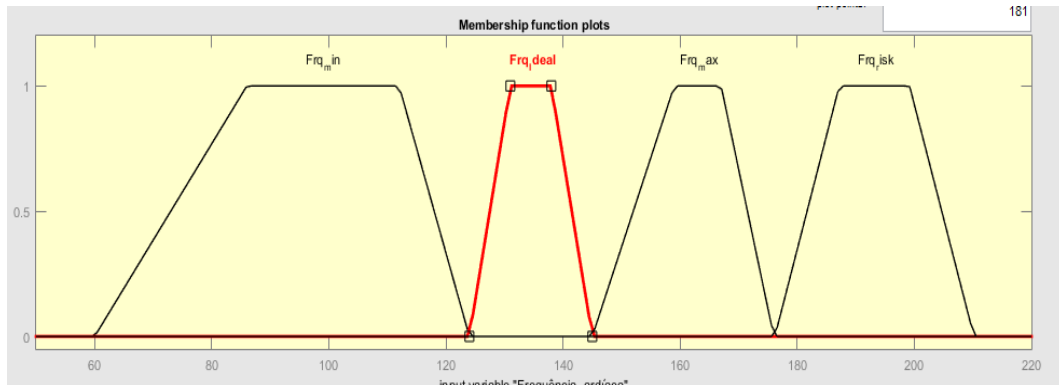
A partir dos dados das Tabelas 3,4,5,6,7 e 8 foram elaborados gráficos para cada indivíduo, cada gráfico mostra as funções de pertinência (entrada e saída, superfície de resultados relacionados as regras de inferência e a saída do controlador).

Em relação ao Usuário 1 (U1), analisam-se os resultados para as frequências cardíacas (FC) e também para as cargas. As duas variáveis são classificadas pelo sistema apoio à decisão como:

- 138 bpm, FC ideal; 90 W, carga leve;
- 179 bpm, FC de risco; 120 W, carga média;
- 170 bpm, FC máxima; 200 W, carga pesada.

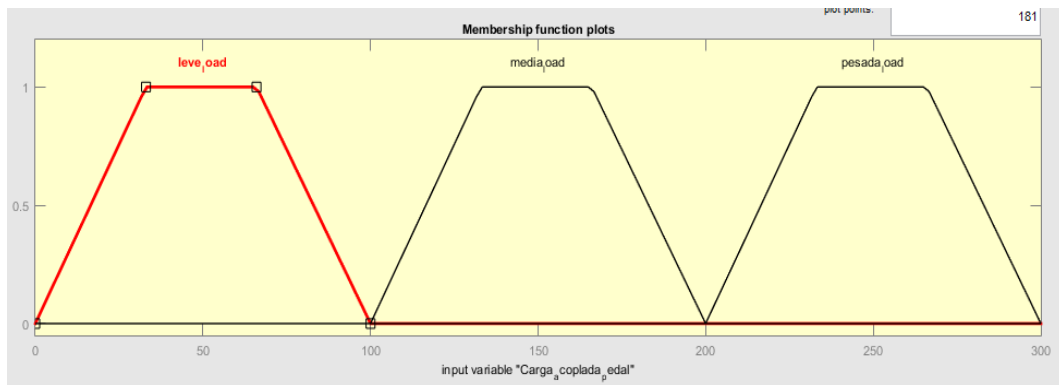
As Figuras de 27 á 33 apresentam os resultados simulados pelo MATLAB® dos dados obtidos do teste desenvolvido pelo usuário U1.

Figura 27: Função de Pertinência da Frequência Cardíaca do usuário U1



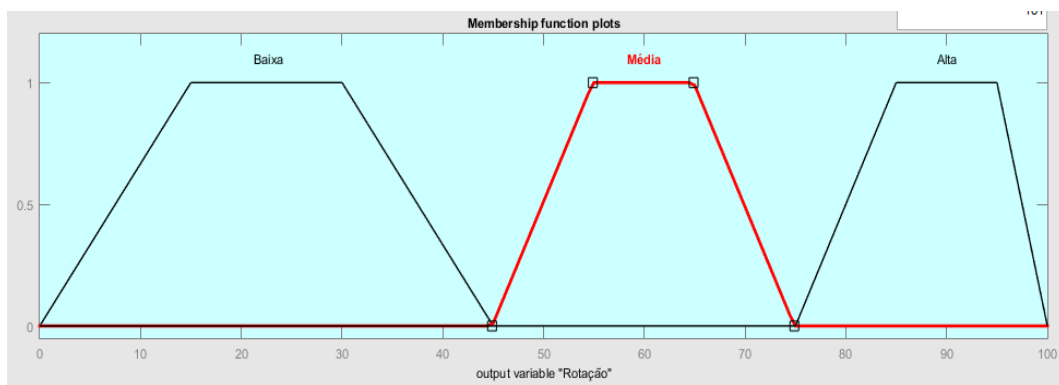
Fonte: autoria própria.

Figura 28 - Função de Pertinência da carga do usuário U1



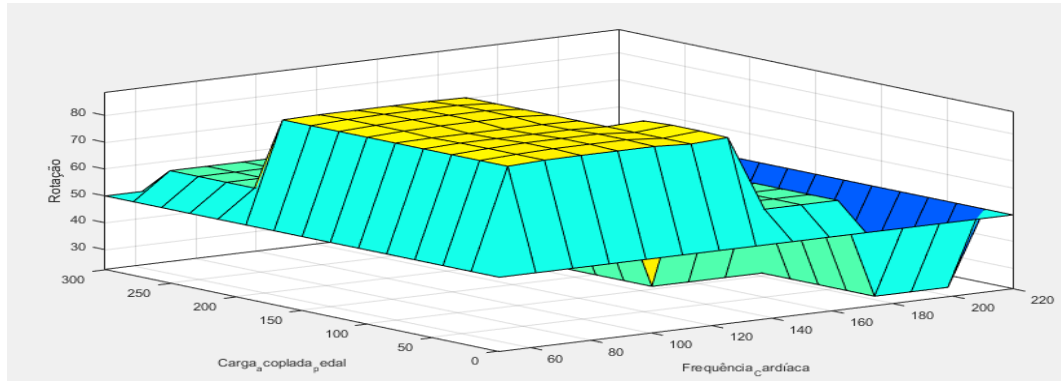
Fonte: autoria própria.

Figura 29 - Função de Pertinência da rotação do usuário U1



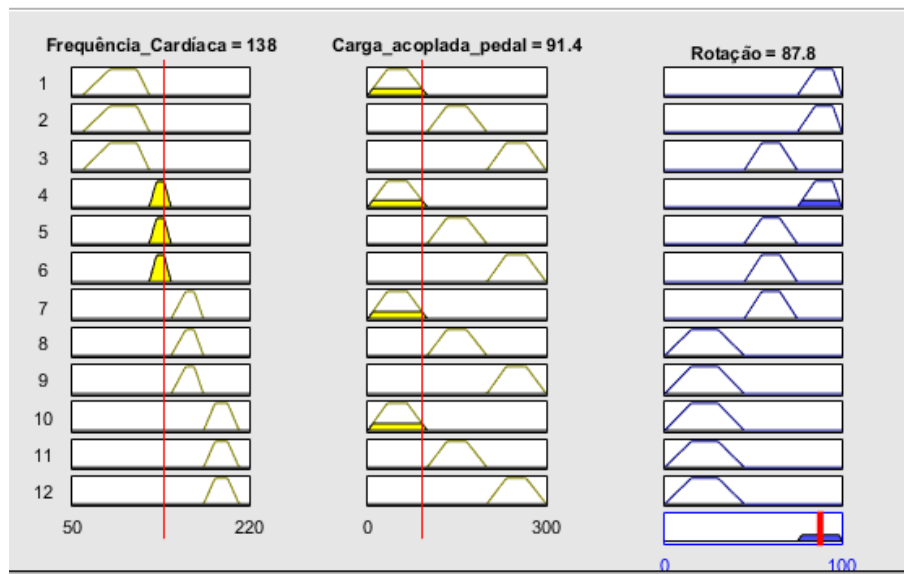
Fonte: autoria própria.

Figura 30 - Superfície gerada para perfil do usuário U1



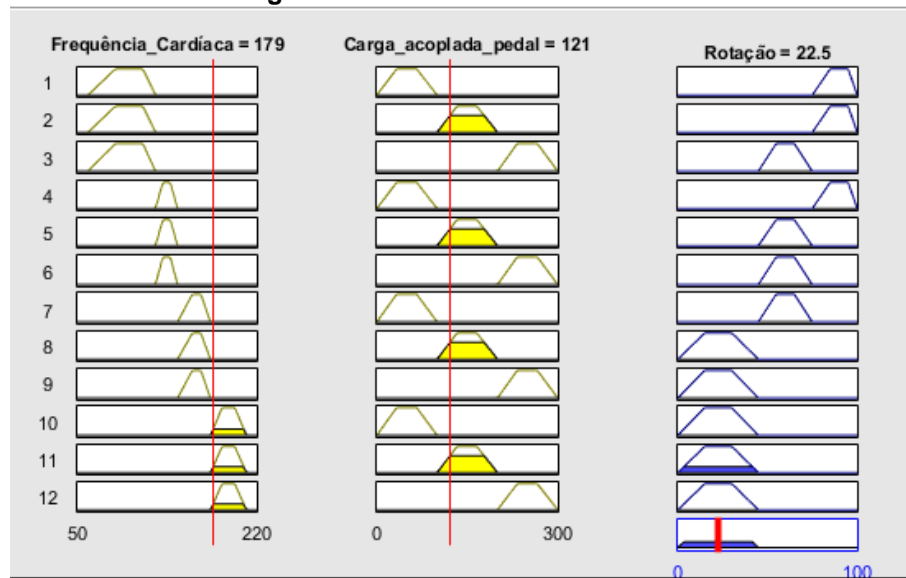
Fonte: autoria própria.

Figura 31- Saída 1 usuário U1



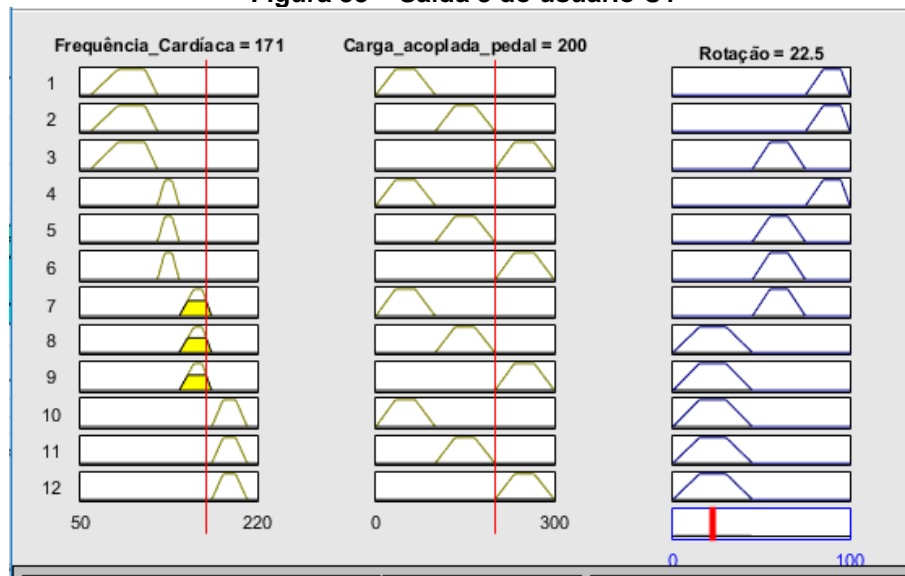
Fonte: autoria própria.

Figura 32- Saída 2 do usuário U1



Fonte: autoria própria.

Figura 33 – Saída 3 do usuário U1



Fonte: autoria própria.

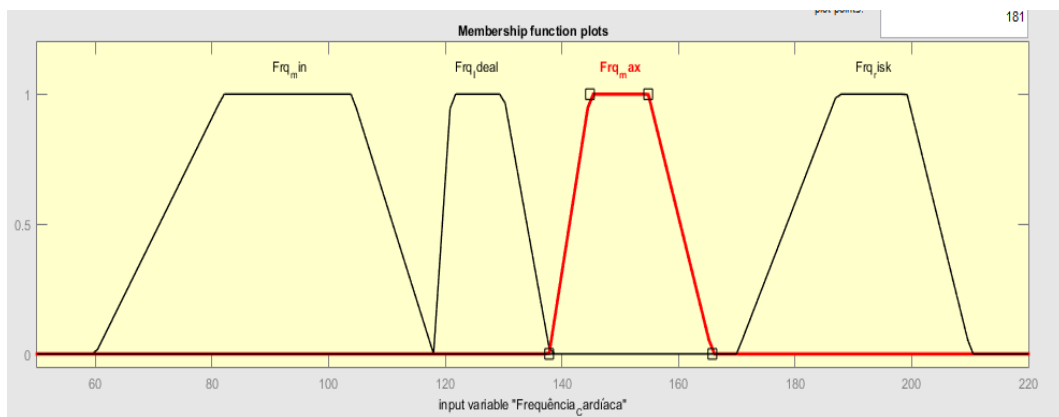
Comparando os gráficos das funções de pertinência geradas pelo MATLAB® das figuras 27 e 28 com o resultado do ARDUINO® (conforme mostram as Tabelas 3 e 6) verificou-se que o sistema de apoio à decisão e de simulação no MATLAB® apresentam o mesmo resultado. Já para a saída 1 figura 31 houve uma diferença de 3,4% no valor da rotação recomendada em relação ao ARDUINO (tabela 6). Na saída 2 teve uma alteração de 11% na rotação recomendada e a saída 3 apresentou a mesma rotação na simulação e no sistema de apoio à decisão.

Com referência ao Usuário 2 (U2), os resultados dos testes para as frequências cardíacas (FC) e também para as cargas foram classificadas pelo sistema apoio à decisão como:

- 129 bpm, FC ideal; 100 W, carga leve;
- 105 bpm, FC ideal;140 W, carga média;
- 128 bpm, FC ideal; 200 W, carga média.

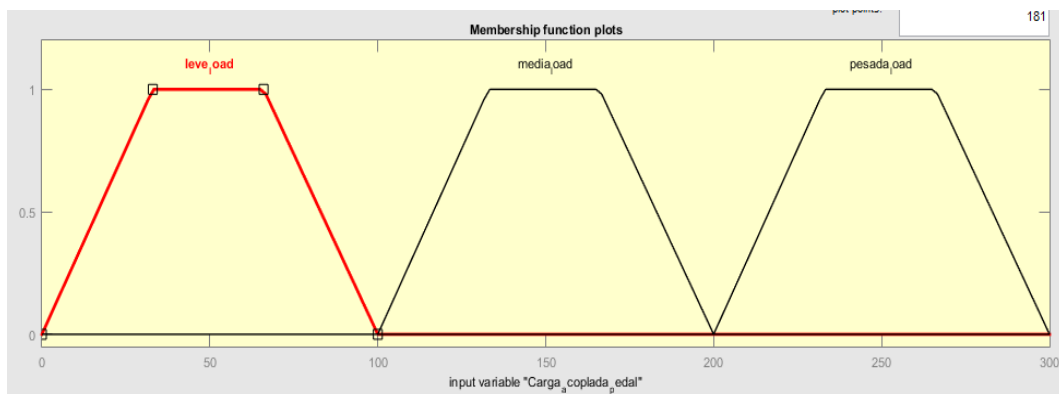
De acordo com as Regras Fuzzy, foram feitas as figuras de 34 á 40 apresentam os resultados simulados pelo MATLAB® dos dados obtidos do teste desenvolvido pelo usuário U2.

Figura 34 - Função de Pertinência da frequência cardíaca do usuário U2



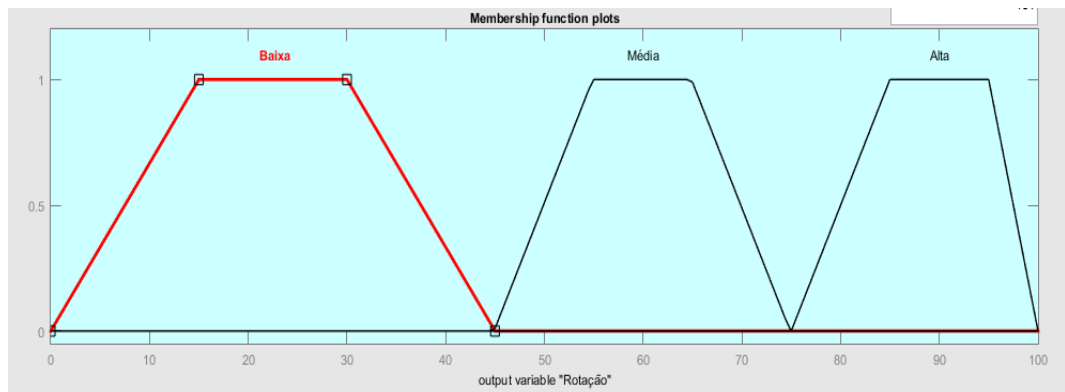
Fonte: autoria própria.

Figura 35 - Função de Pertinência da carga do usuários U2 e U3



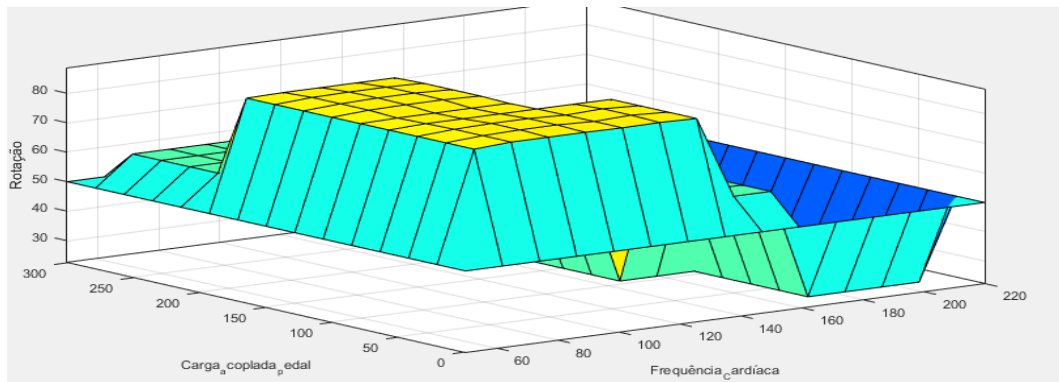
Fonte: autoria própria.

Figura 36 - Função de Pertinência Rotação do usuário U2 e U3



Fonte: autoria própria.

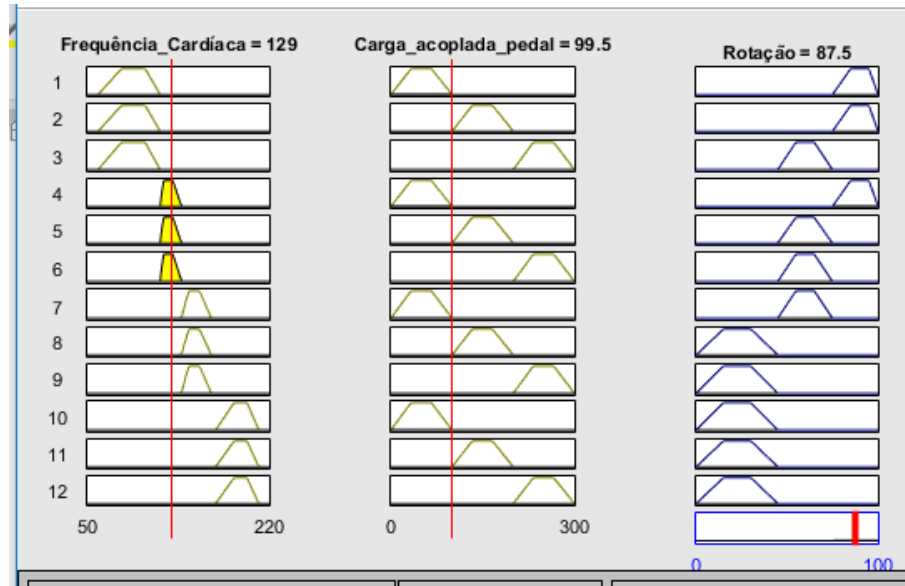
Figura 37 - Superfície gerada para perfil do usuário U2 e U3



Fonte: Autora

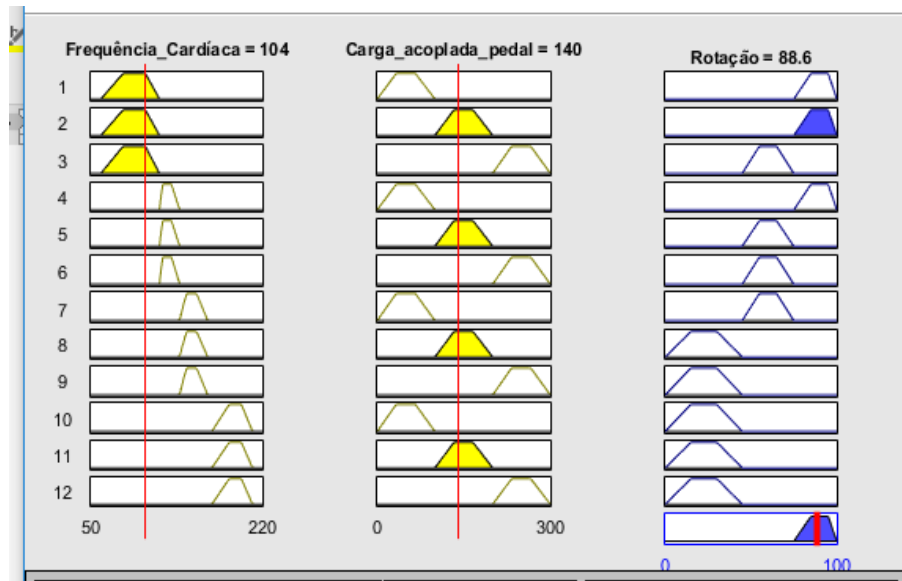
Verificando o gráfico de superfície da figura 37 percebe-se que a medida que a frequência cardíaca e a carga decrescem o grau de pertinência da variável rotação vai aumentando de acordo com o programado pelo especialista.

Figura 38 – Saída 1 do usuário U2



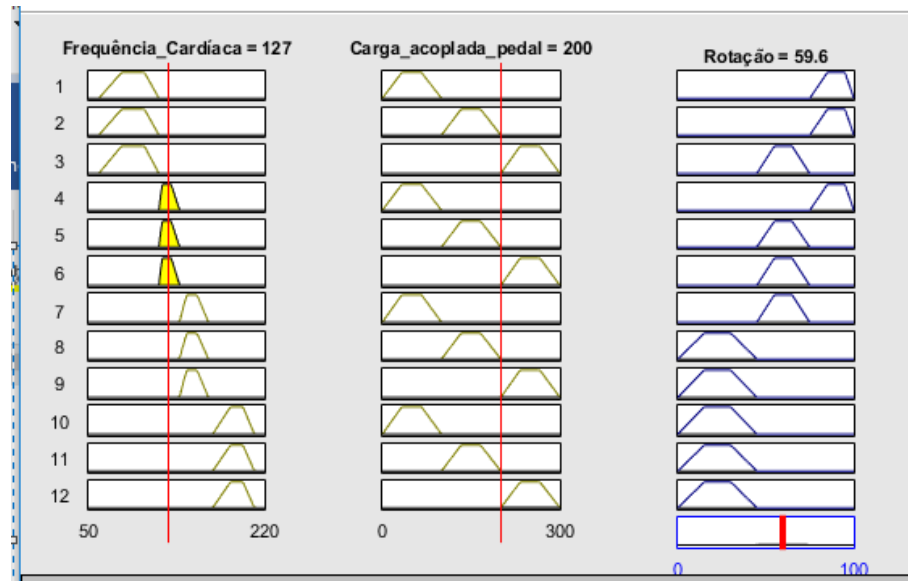
Fonte: autoria própria.

Figura 39 - Saída 2 do usuário U2



Fonte: autoria própria.

Figura 40- Saída 3 do usuário U2



Fonte: autoria própria.

Comparando os gráficos das funções de pertinência geradas pelo MATLAB® das figuras 34 e 35 com o resultado do ARDUINO conforme mostram as tabelas 4 e 7 verificou-se que durante o exercício físico o usuário 2 apresentou a frequência cardíaca ideal para todas as intensidades de carga. Houve variação na recomendação da rotação na saída 2, figura 39, a diferença foi de 4,1 % (tabela 7). Nas saídas 1 e 3 (figuras 38 e 40) a recomendação de rotação é a mesma entre sistema de recomendação e simulação. Foi recomendado que o usuário mantivesse a rotação entre alta e média.

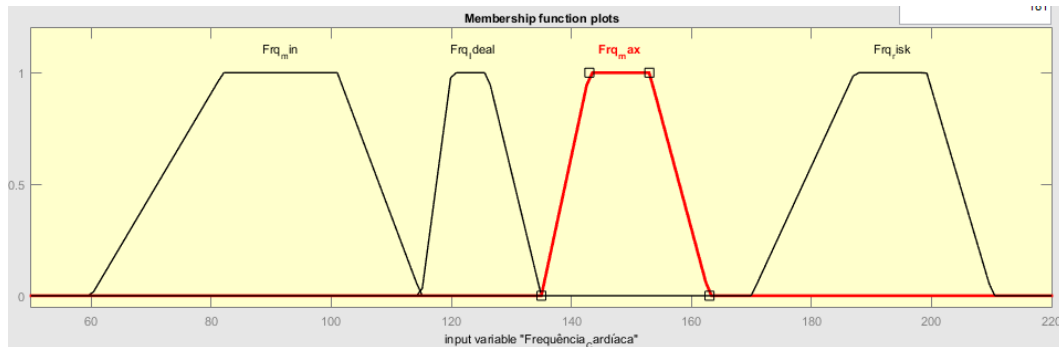
A respeito do usuário 3 (U3), os resultados dos testes para as frequências cardíacas (FC) e para as cargas foram classificadas pelo sistema de recomendação como:

- 152 bpm, FC máxima; 80 W, carga leve;
- 149 bpm, FC máxima; 100 W, carga leve;
- 128 bpm, FC ideal; 150 W, carga média.

As figuras 41, 42, 43 e 44 foram construídas para apresentarem os resultados simulados pelo MATLAB® dos dados obtidos (tabelas 5 e 8) do teste realizado com a plataforma ARDUINO pelo usuário U3. Para as variáveis carga e rotação pode-se utilizar as mesmas funções de pertinência utilizadas para o usuário 2 que são as

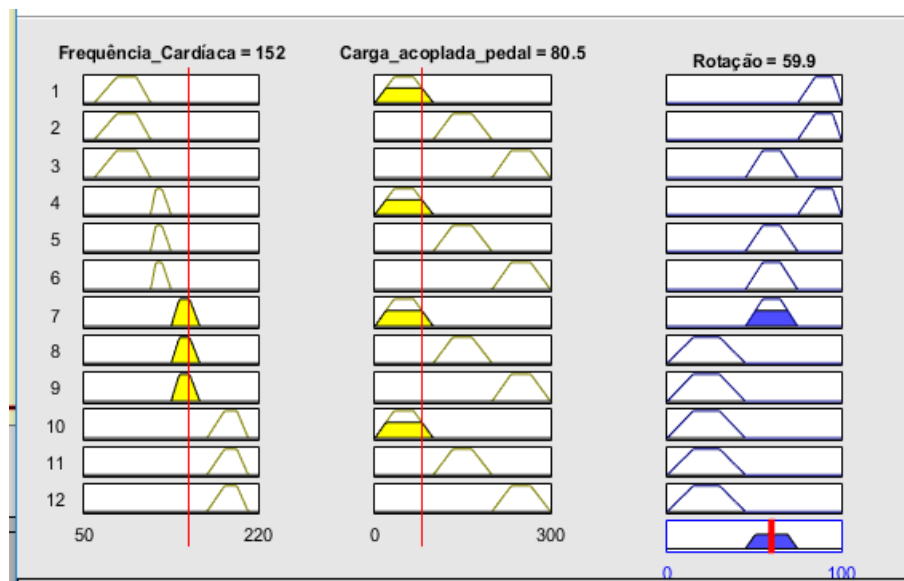
figuras 35 e 36, pois tem as mesmas características. A superfície (figura 37) gerada também é comum com o usuário 2 porque utiliza a mesma base de regras e as funções de pertinência muito são semelhantes.

Figura 41 - Função de Pertinência da frequência cardíaca do usuário U3



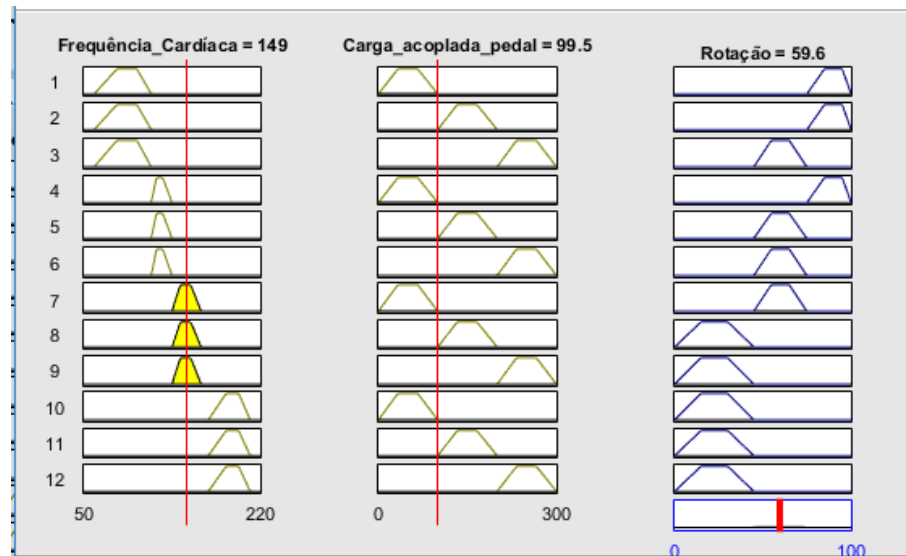
Fonte: autoria própria.

Figura 42 - Saída 1 do usuário U3



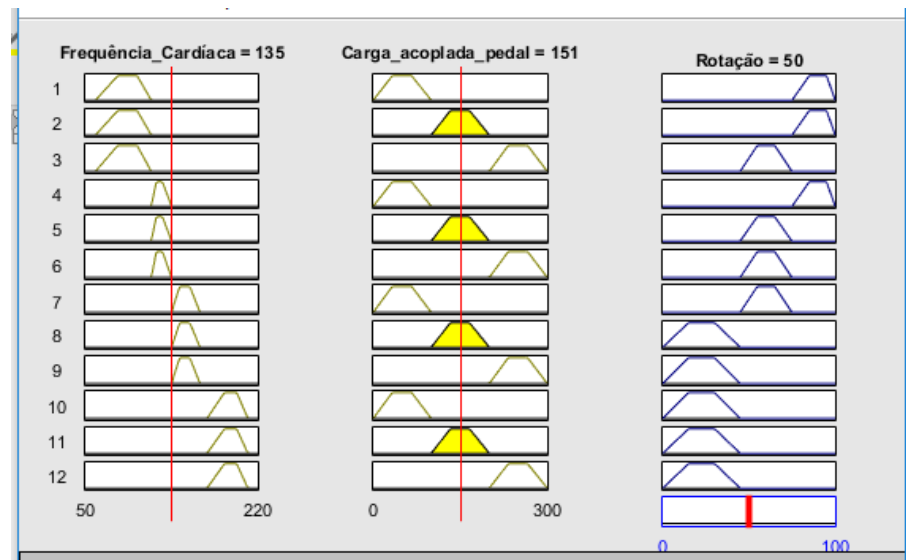
Fonte: autoria própria.

Figura 43 - Saída 2 do usuário U3



Fonte: autoria própria.

Figura 44- Saída 3 do usuário U3



Fonte: autoria própria.

O usuário 3 apresentou a frequência cardíaca máxima para os dois primeiros testes, o sistema de apoio à decisão sugeriu que o usuário pedale com rotação de média. Para a saída 2 figura 43 houve uma diferença de 8,3% no valor da rotação recomendada em relação ao ARDUINO (tabela 8) para as demais saídas a recomendação de rotação foi a mesma.

Após a apresentação e classificação dos dados obtidos nos experimentos dos três usuários as tabelas, 9,10 e 11 mostrarão a comparação entre ARDUÍNO® e o

MATLAB® da etapa de defuzzyficação a qual é o processo de conversão das áreas das figuras, 30, 36 e 42 para um valor numérico para a variável de saída rotação.

Os dados obtidos experimentalmente serão comparados com a simulação do MATLAB® e será calculado o erro da saída do controlador ARDUÍNO® pela equação x a seguir:

$$\text{Erro} = \left(\frac{\text{Valor Matlab} - \text{valor arduino}}{\text{valor Matlab}} \right) * 100 \quad (15)$$

Tabela 9 - Comparação entre a saída do controlador ARDUÍNO® e o MATLAB® para o usuário U1

USUÁRIO	TESTE	ARDUÍNO	MATLAB	ERRO %
1	1	85	88	3,4
1	2	20	22,5	11
1	3	23	22,5	2

Fonte: autoria própria.

Tabela 10 - Comparação entre a saída do controlador ARDUÍNO® e o MATLAB® para os usuários U2

USUÁRIO	TESTE	ARDUÍNO	MATLAB	ERRO %
2	1	87	87	0
2	2	85	88,6	4,1
2	3	60	60	0

Fonte: autoria própria.

Tabela 11 - Comparação entre a saída do controlador ARDUÍNO® e o MATLAB® para o usuário U3

USUÁRIO	TESTE	ARDUÍNO	MATLAB	ERRO %
3	1	60	60	0
3	2	55	59,6	7,7
3	3	50	50	0

Fonte: autoria própria.

As Tabelas 9, 10 e 11 apresentaram os resultados das saídas calculados pelo método centroide pelos softwares ARDUÍNO® e MATLAB® assim como o erro percentual entre eles, resultantes das combinações dos dados dos testes feitos no sistema de microgeração com o cada usuário tanto no ARDUÍNO® quanto no MATLAB®. A saída do sistema é um valor numérico e real que representa a rotação adequada para este usuário.

A análise dos resultados obtidos foi:

Usuário 1: o erro médio foi de 5,46%. Neste caso o valor do erro não irá alterar a recomendação para o usuário 1 que após atingir a frequência cardíaca máxima deverá permanecer em rotação baixa. A possível causa desse erro é devido ao MATLAB® trabalhar com valores decimais enquanto o ARDUÍNO® com números inteiros.

Usuário 2: o erro médio obtido é 1,35% considerado pequeno e não irá alterar a recomendação para o usuário 2. A recomendação média para este usuário foi manter a rotação alta devido ao mesmo ter a frequência cardíaca estável.

Usuário 3: O erro médio foi de 2,58% considerado pequeno e não irá alterar a recomendação para o usuário 3 que é de manter a rotação média.

6 CONCLUSÃO E RECOMENDAÇÕES PARA TRABALHOS FUTUROS

6.1 CONCLUSÕES

O sistema de recomendação construído monitora a frequência cardíaca e recomenda uma rotação adequada de acordo principalmente com a frequência cardíaca e da carga. O circuito eletrônico de controle fez uso de uma plataforma de prototipagem eletrônica, o ARDUINO®, com processador AT Mega 2560 que tornou o circuito de baixo custo, simples, de fácil programação e integração com os sensores e demais dispositivos do sistema

O software de apoio à decisão desenvolvido é uma plataforma de código *open source* usada para a construção de projetos eletrônicos e também se utilizou da biblioteca eFLL para implementar a lógica *Fuzzy* recurso que permitiu elaborar as estratégias de recomendação do sistema.

Devido à demanda por novas fontes alternativas e limpas de energia este trabalho apresentou o projeto que converte a energia mecânica utilizada durante a atividade física em energia elétrica. Para isto, foi acoplado um gerador na polia de transmissão do pedal de uma bicicleta ergométrica, mas para gerar energia limpa utilizando o esforço humano deve-se garantir a segurança do usuário, pois a atividade física de pedalar exige do ciclista um esforço cardíaco grande, sendo assim, foi desenvolvido o sistema de recomendação baseado em lógica *Fuzzy*.

Acredita-se que o sistema desenvolvido neste trabalho atingiu os seus objetivos de gerar energia elétrica pela atividade física de pedalar e de orientar a atividade física, pois mostra em um display a frequência cardíaca, classifica a mesma como mínima, ideal, máximo e de risco e orienta a rotação que o usuário deve pedalar. O protótipo criado poderia ser adotado em academias visto que o mesmo, além de gerar energia, proporciona segurança na prática do exercício físico.

Os resultados práticos obtidos neste trabalho estão de acordo com os dados adquiridos pelo software MATLAB®, que, conforme a comparação com os dados obtidos durante o teste, apresentam um erro médio de 3,1%.

O software e o hardware confeccionados garantem a utilização da bicicleta ergométrica gerando energia, monitorando a frequência cardíaca e recomendando a rotação adequada para a prática de exercícios físicos.

6.2 DIFICULDADES ENCONTRADAS E RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Algumas dificuldades foram identificadas neste trabalho, dentre elas a mais crítica foi obter a frequência cardíaca utilizando sensores de batimento, foram testados alguns modelos antes de achar o mais estável. Também houve a necessidade de implementar filtro na alimentação do sensor de batimento cardíaco devido à interferência. Outro ponto, foi a implementação da lógica *Fuzzy* no ARDUÍNO®, a sua integração com os sensores e fazer o monitoramento em tempo real.

Existem algumas recomendações para trabalhos futuros dentre elas está a instalação de um conversor na saída do gerador para manter estável a potência de saída do mesmo. Acoplar no pedal da bicicleta uma célula de carga para medir a força aplicada pelo ciclista no pedal. Inserir um sensor de corrente para determinar a potência elétrica durante a prática do exercício físico e deixar a bicicleta em um ambiente de academia para verificar a potência média gerada durante o período de funcionamento da mesma. Inserir na interface com o usuário um teclado para digitar a idade deste usuário. Aumentar o número de amostras para efetuar os testes no sistema desenvolvido.

REFERÊNCIAS

AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA (ANEEL), **Novas Regras Para Geração Distribuída Entram em Vigor, 2016**. Disponível em: http://www2.aneel.gov.br/aplicacoes/noticias/Output_Noticias.cfm?Identidade=9086&id_area=90. Acesso em: 05 dez 2016.

AGUADO, A. G.; CANTANHEDE, M. A. **Lógica Fuzzy**. 2010. Disponível em: <https://www3.ufpe.br/geodesia/images/simgeo/papers/91-537-1-PB.pdf> Acesso em: 11 abr 2019.

ALMEIDA, M.; ARAUJO, C. G. **Efeitos do treinamento aeróbico sobre a frequência cardíaca**. Rev Bras Med Esporte. 9. 104-112. (2003).

AMENDOLA, M.; SOUZA, A. L.; BARROS, L. C. **Manual do uso da teoria dos conjuntos Fuzzy no Matlab** r 6.5. Manual apresentado no Ciclo de Palestras/2004, realizado na FEAGRI/UNICAMP. Disponível em: Available from http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1983-30832012000200012&lng=en&nrm=iso. Acesso em 22 mar 2019.
<http://dx.doi.org/10.4025/reveducfis.v23i2.12311>.

ARAÚJO, D. S. M. S. de; ARAUJO, C. G. S de. **Aptidão física, saúde e qualidade de vida relacionada à saúde em adultos**. Rev Bras Med Esporte, Niterói , v. 6, n. 5, p. 194-203, Oct. 2000 . Available from http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1517-86922000000500005&lng=en&nrm=isso. Acesso em 28 jun 2019.
<http://dx.doi.org/10.1590/S1517-86922000000500005>.

ARAÚJO, W P.P.; etal. **Metodologia FMEA-FUZZY aplicada à gestão de indicadores de continuidade individuais de sistemas de distribuição de energia elétrica**. 2008.
Disponível em: <https://repositorio.ufsc.br/handle/123456789/106616>. Acesso em 04 fev 2018.

ANDRADE, M.; JACQUES, M.A. **Estudo comparativo de controladores de Mamdani e Sugeno para controle de tráfego em interseções isoladas**. TRANSPORTES. 16. 10.4237/transportes.v16i2.24 .2009.

ARDUINO - Disponível em <http://www.arduino.cc/>. Acesso em 02 jan 2018.

ASSOCIAÇÃO BRASILEIRA DE ARRITMIAS CARDÍACAS (SOBRAC). **Arritmias Cardíacas e Morte Súbita**. disponível em: www.sobrac.org. Acesso em 20 dez 2018.

BRAAE, M.; RUTHERFORD, D.A. **Selection Of Parameters For A Fuzzy LogicController**. Control Systems Centre, University of Manchester, Institute of Science and Technology, Manchester, 1979.

CALHEIROS, D.M. **Microgeração de Energia Elétrica através do Exercício Físico**. 2016. 76 f. Dissertação de Mestrado (Mestrado em Engenharia do Ambiente) — Faculdade de Engenharia da Universidade do Porto 2016.

CAPUTO, E.L.; SILVA, M. C.; ROMBALDI, A. J. **Comparison of maximal heart rate attained by distinct methods**. Rev. educ. fis. UEM, Maringá , v. 23, n. 2, p. 277-284, 2012.

CHALERMTHAI, B.; et al. **Recovery of Useful Energy from Lost Human Power in Gymnasium**. *Department of Electrical Engineering and Computer Science, Masdar Institute of Science and Technology Abu Dhabi, United Arab Emirates , 2015.

COSTA, R. C.; PRATES, C. P. T. **“O papel das Fontes Renováveis de Energia no Sector Energético e Barreiras à sua Penetração no Mercado”**, Departamento de Gás, Petróleo, Co-Geração e Outras Fontes de Energia, BNDES, Rio de Janeiro, n. 21, p. 5-30, mar. de 2005.

DE LIZ, M. B. **Contribuição para a Redução da Interferência Eletromagnética em Fontes chaveadas**, 2003. 178 f. Tese de (Doutorado em Engenharia) Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, 2003.

DEL TORO, V. **Fundamentos de Máquinas Elétricas**. 1ª Ed. Rio de Janeiro: LTC, 1994.

DIAS, L. R.; LIMA, G. B.; RODRIGUES, D B. **Sistema de Geração de Energia Elétrica a partir de uma Bicicleta Ergométrica**. Escola de Engenharia de São Carlos (EESC) – Universidade de São Paulo (USP), São Carlos – SP. Instituto de Ciências Tecnológicas e Exatas (ICTE), Universidade Federal do Triângulo Mineiro (UFTM), Uberaba – MG, Out. de 2016.

DIAS, M.; LIMA, J. R.; NOVAES, J. S.; **Cadência de pedalada no ciclismo: uma revisão de literatura. Motricidade** 3(1): 270-278. Rio de Janeiro RJ, dezembro de 2006.

ELENO, T. G.; KOKUBUN, E. **Sobrecarga fisiológica do dribble no handebol: um estudo pelo lactato sanguíneo e frequência cardíaca em sujeitos treinado e não-treinados**. Revista de Educação Física/UEM, Maringá, v.13, n.1, p. 109-114. 2002.

FACHIN, O. **Fundamentos de metodologia**. 6ªed São Paulo: Saraiva, 2005.

GIL, A.C ; Como Elaborar Projetos de Pesquisa. 4a ed. São Paulo: Editora Atlas S/A, 2002.

GILMORE, A. M. **Human Power: Energy Recovery from Recreational Activity**. *Guelph Engineering Journal*, (1), 8 - 16. ISSN: 1916-1107. ©2008.

GOMIDE, F. A. C.; GUDWIN, R. R; TANSCHER, R. **Conceitos Fundamentais Da Teoria De Conjuntos Fuzzy, Lógica Fuzzy E Aplicações**.

Universidade Estadual de Campinas (UNICAMP). Campinas, 2012.

IVANQUI, J. **Esteira eletrônica com velocidade controlada por lógica fuzzy**. 2005. 105 f. Dissertação (Mestrado em Engenharia Elétrica e Informática Industrial) – Universidade Tecnológica Federal do Paraná, Curitiba, 2005.

KUMRU, M.; KUMRU, P. Y. **Fuzzy FMEA application to improve purchasing process in a public hospital**. *Applied Soft Computing*, v. 13, n. 1, p. 721-733, 2013.

LA GERCHE, A. **Can intense endurance exercise cause myocardial damage and fibrosis?**. *Current Sports Medicine Reports*. 2013 Mar-Apr;12(2):63-9.

LEE, C.C. **Fuzzy Logic in Control Systems: Fuzzy Logic Controller**. *IEEE Transactions on Systems, Man, and Cybernetics*. vol. 20. no. 2. -april, 1990.

MACAGNAN, A L.; NETO, J. A R.; NETO, R. N. **Determinação da Eficiência Energética da Tração Humana na Geração de Energia Elétrica Através de Uma Bicicleta Ergométrica**. 2016. 87f. Trabalho de Conclusão de Curso – Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

MARINS, J. C .B.; et al . **Validação do tempo de mensuração da frequência cardíaca após esforço submáximo a 50 e 80%**. *Rev Bras Med Esporte*, Niterói , v. 4, n. 4, p. 114-119, Aug. 1998 . Available from http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1517-69219980004&lng=en&nrm=iso. access on 06 July 2019. <http://dx.doi.org/10.1590/S1517-86921998000400004>.

MARON, B. J.; PELLICCIA, A. **Cardiac Remodeling and the Risks of Sports, Including Sudden Death**. *Circulation*. Oct-2006;114:1633–1644. access on 10 July 2019. https://www.ahajournals.org/doi/full/10.1161/CIRCULATIONAHA.106.613562?url_ver=Z39.88-2003&rfr_id=ori:rid:crossref.org&rfr_dat=cr_pub%3dpubmed

MORAES, C. C.; CASTRUCCI, P. L. **Engenharia de Automação Industrial**. 4 ed. Rio de Janeiro: LTC, 2001.

NASCIMENTO JUNIOR, G. C. **Máquinas Elétricas: teoria e ensaios**. 2. ed. São Paulo: Érica, 2007.

NASCIMENTO, C. L.; YONEYAMA, T. **Inteligência Artificial em Controle e Automação**. 1a. ed. São Paulo: Editora Edgard Blücher e FAPESP, ISBN 85-212-0249-0 e 85-212-0310-1, Julho 2000.

O'KEEFE, J.H.; et al. **A. Potential Adverse Cardiovascular Effects From Excessive Endurance Exercise**. *Mayo Clinical Proceedings*. 2012 Jun; 87(6): 587–595.

PRODANOV, C. C.; DE FREITAS, E. C. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico-2ª Edição**. Editora Feevale, 2013.

PRUDENTE, M.; PFITSCHER, L. L.; EMMENDOERFER, G.; ROMANELI, E. GULES, R. **Voltage Multiplier Cells Applied to Non-Isolated DC-DC Converters**. IEEE TRANSACTIONS ON POWER ELECTRONICS, VOL. 23, NO. 2, MARCH 2008

RANIERI, F. **Sistema Supervisório de Parâmetros de Máquinas Elétricas via TC/IP e Painel eletrônico de Mensagens**. 2007. 180 f. Dissertação de (Mestrado em Engenharia) Programa de Pós Graduação em Engenharia Elétrica, Universidade de São Paulo, São Paulo, 2007;

SAMPAIO, L. P. **“Uma Nova Metodologia de Projeto e Controle para o Inversor Boost (CSI) Monofásico, para o Aproveitamento de Fontes Alternativas e Renováveis de Energia Elétrica”**. 2010. 110 f. Tese (Doutorado em Engenharia Elétrica) — Faculdade de Engenharia, Universidade Estadual Paulista - UNESP, Ilha Soleira, 2010.

SANDRI, S.; CORREA, C. **Lógica Nebulosa**. INPE, ITA São José dos Campos – São Paulo, 1999.

SANTOS, I. E dos. **Textos Selecionados de Métodos e Técnicas de Pesquisa Científica**. Rio de Janeiro: Impetus, 2003

SEIDL, Jamie; ASPLUND, Chad. Effects of Excessive Endurance Activity on the Heart. Current Sports Medicine Reports. **13(6):361–364, NOV 2014**

SHAW, I. S; SIMÕES, M.G. **Controle e Modelagem Fuzzy**. 2ª ed. São Paulo: Edgard Blücher Ltda, 2007.

SILVA, E. L; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. UFSC, Florianópolis, 4a. edição, 2005.

SILVEIRA, P. R. dos; SANTOS, W. E. dos. **Automação e Controle Discreto**. 3 ed. São Paulo: Érica, 2001.

SOFTWARE LIVRE BRASIL. AJ ALVES: eFLL – **Uma Biblioteca Fuzzy para Arduino e Sistemas Embarcados**. Disponível em: <http://softwarelivre.org/psl-pi/blog/aj-alves-efll---uma-biblioteca-fuzzy-para-arduino-e-sistemas-embarcados>
Acesso em: 29 dez 2018.

TAVARES, C. A. P. **Estudo Comparativo de Controladores Fuzzy Aplicados a um Sistema Solar Fotovoltaico**. 2009. 129f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, 2009.

APÊNDICE A – CÓDIGO DE PROGRAMAÇÃO DO SISTEMA DE MONITORAMENTO E RECOMENDAÇÃO

```

//#define SHOW_RAW_VALUE
#define SHOW_MEAN_VALUE

#include <math.h>
#include <Fuzzy.h>
#include <FuzzyComposition.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzyOutput.h>
#include <FuzzyRule.h>
#include <FuzzyRuleAntecedent.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzySet.h>
#include <LiquidCrystal.h>
#include <EEPROM.h>
#include <string.h>

// HEART SENSOR VARIABLES -----
#define HEART_PIN 0
#define HEART_SIZE 5
#define HEART_THRESHOLD 320
#define HEART_MEANSIZE 20

unsigned long timer, heart_peakTime[HEART_SIZE];
unsigned long heart_timer = 0;
bool heart_aboveThreshold = false;
int heart_index = 0;
unsigned int BPM;
int heart_mean[HEART_MEANSIZE];

```



```

int heart_meanIndex = 0;
// -----

// HALL SENSOR VARIABLES (DIGITAL) -----
---
#define HALL_PIN 5
float FREQ_HALL;
unsigned long last_pass = 0;
bool hall_peaked = false;
// -----

// BUTTON VARIABLES -----

#define BUTTON1_PIN 4
#define BUTTON2_PIN 3
#define BUTTON_SAMPLES 1
unsigned long button_timer = 0;
char button1_counter = 0;
char button2_counter = 0;
char BUTTON1 = 0;
char BUTTON2 = 0;

// MODE & USER VARIABLES -----
#define MODE_USER_CHOICE 0
#define MODE_PEDALING 1
#define MODE_DATA_ACQUISITION 2
char MODE = MODE_USER_CHOICE;
char USER = 1;

// LOAD CHOICE VARIABLES -----
#define LOAD_CHOICE_ANALOG_PIN 1
int load_choice= ( (100) * (FREQ_HALL / 100));

int Pinload_3 = 21;

```

```
int Pinload_2 = 25;
int Pinload_1 = 23;

// -----

// AVERAGE VARIABLES -----

double AVG_BPM = 0;
double AVG_LOAD = 0;
unsigned long AVG_N = 0;
unsigned long avg_timer = 0;

// -----

// PEDALING DETECTION -----

unsigned long pedal_timer = 0;
int PEDALING = 0;
unsigned long start_pedaling_time = 0;
unsigned long stop_pedaling_time = 0;

// -----

// SCREEN -----

#define N_SCREEN 3
int SCREEN = 0;

// -----

// FUZZY VARIABLES (AND SETS) -----
Fuzzy* fuzzy = new Fuzzy();
float output;
```

```
// Frecuencia cardiaca
```

```
FuzzySet* frqcarmin20 = new FuzzySet(60, 86, 112, 124);
```

```
FuzzySet* frqcarideal20 = new FuzzySet(124, 131, 138, 145);
```

```
FuzzySet* frqcarmax20 = new FuzzySet(145, 159, 167, 176);
```

```
FuzzySet* frqcarrisk = new FuzzySet(176, 187, 199, 210);
```

```
// carga
```

```
FuzzySet* leve = new FuzzySet(0, 33, 66, 100); // carga leve
```

```
FuzzySet* media = new FuzzySet(100, 133, 166, 200); // carga ideal
```

```
FuzzySet* pesada = new FuzzySet(200, 233, 266, 300); // carga pesada
```

```
// verificar carga
```

```
// -----
```

```
// LCD VARIABLES -----
```

```
unsigned long LCD_timer = 0;
```

```
// LiquidCrystal(rs, enable, d4, d5, d6, d7);
```

```
LiquidCrystal LCD(13, 12, 11, 10, 9, 8);
```

```
// -----
```

```
void setup() {
```

```
  Serial.begin(115200);
```

```
  // SENSOR SETUP
```

```
  pinMode(HALL_PIN, INPUT);
```

```
  pinMode(BUTTON1_PIN, INPUT);
```

```
  pinMode(BUTTON2_PIN, INPUT);
```

```
  digitalWrite(BUTTON1_PIN, HIGH);
```

```
  digitalWrite(BUTTON2_PIN, HIGH);
```

```
// LOAD OUTPUT SETUP
```

```
pinMode (Pinload_3, OUTPUT);  
pinMode (Pinload_2, OUTPUT);  
pinMode (Pinload_1, OUTPUT);
```

```
// LCD SETUP
```

```
LCD.begin(16, 2);
```

```
// FUZZY SETUP
```

```
// FuzzyInput frequencia
```

```
FuzzyInput* BPM = new FuzzyInput(1);  
BPM->addFuzzySet(frqcamin20);  
BPM->addFuzzySet(frqcarideal20);  
BPM->addFuzzySet(frqcarmax20);  
BPM->addFuzzySet(frqcarrisk);  
fuzzy->addFuzzyInput(BPM);
```

```
// FuzzyInput carga
```

```
FuzzyInput* load_choice = new FuzzyInput(3);  
load_choice->addFuzzySet(leve);  
load_choice->addFuzzySet(media);  
load_choice->addFuzzySet(pesada);  
fuzzy->addFuzzyInput(load_choice);
```

```
// Fuzzyoutput Rotação
```

```
FuzzyOutput* S = new FuzzyOutput(1);
```

```

FuzzySet* A = new FuzzySet(75,85,95,100);           //definindo parametros de
severidade
S->addFuzzySet(A);
FuzzySet* Ma = new FuzzySet(45,55, 65,75);
S->addFuzzySet(Ma);
FuzzySet* D = new FuzzySet(0,15,30,45);
S->addFuzzySet(D);
FuzzySet* risk= new FuzzySet(90, 92, 100, 100);
S->addFuzzySet(risk);

fuzzy->addFuzzyOutput(S);

// Building FuzzyRule 1

FuzzyRuleAntecedent* iffrqcarmin20ANDleve = new FuzzyRuleAntecedent();
iffrqcarmin20ANDleve->joinWithAND(frqcarmin20, leve);
FuzzyRuleConsequent* thenSisA = new FuzzyRuleConsequent();
thenSisA ->addOutput(A);

FuzzyRule* fuzzyRule1 = new FuzzyRule(1, iffrqcarmin20ANDleve, thenSisA);
fuzzy->addFuzzyRule(fuzzyRule1);

// Building FuzzyRule 2

FuzzyRuleAntecedent* iffrqcarmin20ANDmedia= new FuzzyRuleAntecedent();
iffrqcarmin20ANDmedia->joinWithAND(frqcarmin20, media);
thenSisA->addOutput(A);
FuzzyRule* fuzzyRule2 = new FuzzyRule(2, iffrqcarmin20ANDmedia, thenSisA);
fuzzy->addFuzzyRule(fuzzyRule2);

// Building FuzzyRule 3

FuzzyRuleAntecedent* iffrqcarmin20ANDpesada = new FuzzyRuleAntecedent();
iffrqcarmin20ANDpesada->joinWithAND(frqcarmin20, pesada);

```

```

FuzzyRuleConsequent* thenSisMa = new FuzzyRuleConsequent();
thenSisA->addOutput(Ma);
FuzzyRule* fuzzyRule3 = new FuzzyRule(3, iffrqcarmin20ANDpesada, thenSisMa);
fuzzy->addFuzzyRule(fuzzyRule3);

```

```
// Building FuzzyRule 4
```

```

FuzzyRuleAntecedent* iffrqcarideal20ANDleve = new FuzzyRuleAntecedent();
iffrqcarideal20ANDleve->joinWithAND(frqcarideal20, leve);
// FuzzyRuleConsequent* thenSisA = new FuzzyRuleConsequent();
thenSisA ->addOutput(A);
FuzzyRule* fuzzyRule4 = new FuzzyRule(4, iffrqcarideal20ANDleve, thenSisA);
fuzzy->addFuzzyRule(fuzzyRule4);

```

```
// Building FuzzyRule 5
```

```

FuzzyRuleAntecedent* iffrqcarideal20ANDmedia= new FuzzyRuleAntecedent();
iffrqcarideal20ANDmedia->joinWithAND (frqcarideal20, media);
// FuzzyRuleConsequent* thenSisMa = new FuzzyRuleConsequent();
thenSisMa->addOutput(Ma);

```

```

FuzzyRule* fuzzyRule5 = new FuzzyRule(5, iffrqcarideal20ANDmedia, thenSisMa);
fuzzy->addFuzzyRule(fuzzyRule5);

```

```
// Building FuzzyRule 6
```

```

FuzzyRuleAntecedent* iffrqcarideal20ANDpesada= new FuzzyRuleAntecedent();
iffrqcarideal20ANDpesada->joinWithAND (frqcarideal20, pesada);
FuzzyRuleConsequent* thenSisD = new FuzzyRuleConsequent();
thenSisD->addOutput(D);

```

```

FuzzyRule* fuzzyRule6 = new FuzzyRule(6, iffrqcarideal20ANDpesada, thenSisD);
fuzzy->addFuzzyRule(fuzzyRule6);

```

```
// Building FuzzyRule 7
```

```
FuzzyRuleAntecedent* iffrqcarmax20ANDpesada= new FuzzyRuleAntecedent();
iffrqcarmax20ANDpesada->joinWithAND (frqcarmax20, pesada);
//FuzzyRuleConsequent* thenSisD = new FuzzyRuleConsequent();
thenSisD->addOutput(D);
```

```
FuzzyRule* fuzzyRule7 = new FuzzyRule(7, iffrqcarmax20ANDpesada, thenSisD);
fuzzy->addFuzzyRule(fuzzyRule7);
```

```
// Building FuzzyRule 8
```

```
FuzzyRuleAntecedent* iffrqcarmax20ANDmedia= new FuzzyRuleAntecedent();
// iffrqcarmax20ANDmediaconsequent* thenSisD = new FuzzyRuleConsequent();
thenSisD->addOutput(D);
FuzzyRule* fuzzyRule8 = new FuzzyRule(8, iffrqcarmax20ANDmedia, thenSisD);
fuzzy->addFuzzyRule(fuzzyRule8);
```

```
// Building FuzzyRule 9
```

```
FuzzyRuleAntecedent* iffrqcarmax20ANDleve= new FuzzyRuleAntecedent();
iffrqcarmax20ANDleve->joinWithAND (frqcarmax20, leve);
// FuzzyRuleConsequent* thenSisMa = new FuzzyRuleConsequent();
thenSisMa->addOutput(Ma);
FuzzyRule* fuzzyRule9 = new FuzzyRule(9, iffrqcarmax20ANDmedia, thenSisMa);
fuzzy->addFuzzyRule(fuzzyRule9);
```

```
// Building FuzzyRule 10
```

```
FuzzyRuleAntecedent* iffrqarriskANDrisk = new FuzzyRuleAntecedent();
iffrqarriskANDrisk ->joinWithAND(frqarrisk, risk);
FuzzyRuleConsequent* thenSisrisk = new FuzzyRuleConsequent();
thenSisrisk->addOutput(risk);
```

```

FuzzyRule* fuzzyRule10 = new FuzzyRule(10, iffrqcarriskANDrisk, thenSisrisk);
fuzzy->addFuzzyRule(fuzzyRule10);
}

```

```

void EEPROM_StringWrite(char *str){
    int i, addr;

    // find last byte ZERO
    for(addr=0; (addr < EEPROM.length()) && (EEPROM.read(addr) != 0); addr++);

    for(i=0; (str[i] != 0) && ( (addr+i) < EEPROM.length()); i++){
        EEPROM.write(addr + i, str[i]);
    }

    // rewrite the ZERO
    EEPROM.write(addr + i, 0);
}

```

```

// HEART SENSOR CODE -----
void calculateBPM() {
    int i;
    double sum = 0;
    for (i = 1; i < HEART_SIZE; i++) {
        int diff = abs(heart_peakTime[i] - heart_peakTime[i - 1]);
        //Serial.print("diff[");
        //Serial.print(i);
        //Serial.print("]:");
        //Serial.println(diff);
        sum += diff;
    }
    BPM = (int)round( 60000.0 / (sum / (HEART_SIZE - 1)) );
    // Serial.print("BPM:");
    //Serial.println(BPM);
}

```



```

double calculateHeartMean() {
    int i;
    double sum = 0;
    for (i = 0; i < HEART_MEANSIZE; i++) {
        sum += heart_mean[i];
    }
    return sum / HEART_MEANSIZE;
}

void avaliateHeart(unsigned long timer, int heart_value) {

    if (heart_value > HEART_THRESHOLD) {
        if (!heart_aboveThreshold) {
            heart_peakTime[heart_index++] = timer;
            //Serial.print("peak[");
            //Serial.print(heart_index-1);
            //Serial.print("]:");
            //Serial.println(timer);
            if (heart_index == HEART_SIZE) {
                heart_index = 0;
                calculateBPM();
            }
        }
        heart_aboveThreshold = true;
    }
    else {
        heart_aboveThreshold = false;
    }

}

// -----

```

```
void loop() {

    timer = millis();

    // BUTTON DETECTION -----
    if( (timer - button_timer) > 20 ){
        button_timer = timer;

        if(digitalRead(BUTTON1_PIN) == 0){

            button1_counter++;
            if(button1_counter == BUTTON_SAMPLES){
                BUTTON1 = 1;
            }
            else if(button1_counter > BUTTON_SAMPLES){
                button1_counter = BUTTON_SAMPLES;
                BUTTON1 = 0;
            }
        }
        else{
            button1_counter = 0;
            BUTTON1 = 0;
        }
    }

    if(digitalRead(BUTTON2_PIN) == 0){

        button2_counter++;
        if(button2_counter == BUTTON_SAMPLES){
            BUTTON2 = 1;
        }
        else if(button2_counter > BUTTON_SAMPLES){
            button2_counter = BUTTON_SAMPLES;
            BUTTON2 = 0;
        }
    }
}
```

```

}
else{
  button2_counter = 0;
  BUTTON2 = 0;
}

}

// MODES -----

if(MODE == MODE_USER_CHOICE){

  if(BUTTON2){
    BUTTON2 = 0;
    MODE = MODE_PEDALING;
  }

  if(BUTTON1){
    BUTTON1 = 0;
    USER++;
    if(USER == 11) USER = 1;
  }

  if( (timer - LCD_timer) > 200 ){
    LCD_timer = timer;

    LCD.setCursor(0, 0);
    LCD.print("USUARIO ");
    LCD.print((int)USER);
    LCD.print("      ");
    LCD.setCursor(0, 1);
    LCD.print("              ");
  }
}

```

```

else if(MODE == MODE_PEDALING){

    // MODE CHANGING
    if(BUTTON2){
        BUTTON2 = 0;
        MODE = MODE_DATA_ACQUISITION;
    }

    // SCREEN CHOICE
    if(BUTTON1){
        BUTTON1 = 0;
        SCREEN++;
        if(SCREEN == N_SCREEN){
            SCREEN = 0;
        }
    }

    // LOAD CHOICE SELECTION -----
    int analogValue = analogRead(LOAD_CHOICE_ANALOG_PIN);
    load_choice      = ( (100) * (FREQ_HALL / 100)) +
100*(floor(analogValue/341.3))*(FREQ_HALL / 100);

    // HEART SENSOR DATA ACQUISITION -----
    if ((timer - heart_timer) > 1) {
        heart_timer = timer;

        int value = analogRead(HEART_PIN);
        heart_mean[heart_meanIndex++] = value;

#ifdef SHOW_RAW_VALUE
        Serial.println(value);
#endif

        if (heart_meanIndex == HEART_MEANSIZE) {

```

```

heart_meanIndex = 0;
double meanValue = calculateHeartMean();
avaliateHeart(timer, meanValue);

#ifdef SHOW_MEAN_VALUE
Serial.println(meanValue);
#endif
}
}
// -----
// HALL SENSOR DATA ACQUISITION -----
if ( (digitalRead(HALL_PIN) == 0) ) {
  if (!hall_peaked) {
    FREQ_HALL = 60000.0 / (timer - last_pass);
    last_pass = timer;
    //Serial.print("HALL:");
    //Serial.println(FREQ_HALL);
  }
  hall_peaked = true;
}
else {
  hall_peaked = false;
}
// -----
// PEDALING DETECTION
if( (timer - pedal_timer) > 50 ){

  unsigned long max_pedal_time;

  pedal_timer = timer;

  switch(load_choice){
    case 300: max_pedal_time = 3000; break;
    case 200: max_pedal_time = 2000; break;

```

```

    case 100: max_pedal_time = 1000; break;
}

if( (timer - last_pass) >= max_pedal_time ){
    if(PEDALING){
        // STOPPED PEDALING
        // USER;AVG_BPM;AVG_LOAD\n
        stop_pedaling_time = timer;
        double delta_t = ((start_pedaling_time - stop_pedaling_time)/60000.0);
        String str = String((int)USER + (String);" + AVG_BPM + (String);" +
AVG_LOAD + (String);" + delta_t + (String)"\n");
        EEPROM_StringWrite(str.c_str());
    }
    PEDALING = 0;
}
else{
    if(!PEDALING){
        // STARTED PEDALING
        AVG_BPM = 0;
        AVG_LOAD = 0;
        AVG_N = 0;
        start_pedaling_time = timer;
    }
    PEDALING = 1;
}
}

// AVG UPDATING
if( (timer - avg_timer) > 500 ){
    avg_timer = timer;

    if(PEDALING){
        unsigned long n_plus = AVG_N + 1;
        AVG_BPM = ( (AVG_BPM*AVG_N) + BPM )/n_plus;
    }
}

```

```

    AVG_LOAD = ( (AVG_LOAD*AVG_N) + load_choice)/n_plus;
    AVG_N = n_plus;
}
}

// FUZZY LOOP -----

//Variavel que recebera os dados de leitura

fuzzy->setInput(1, BPM);           //Adicionando variaveis de entrada
fuzzy->setInput(2, FREQ_HALL);
fuzzy->setInput(3, load_choice);
fuzzy->fuzzify();                 //mandando para o processo de fuzzyficacao

output = fuzzy->defuzzify(1);     // puxando a variavel de saida defuzzyficada

output = fuzzy->defuzzify(1);     // puxando a variavel de saida defuzzyficada

if (load_choice <= 300){
    digitalWrite(Pinload_3 , HIGH);
    digitalWrite(Pinload_2 , HIGH);
    digitalWrite(Pinload_1 , HIGH);
}
if (load_choice <= 200) {
    digitalWrite(Pinload_2 , HIGH);
    digitalWrite(Pinload_1 , HIGH);
    digitalWrite(Pinload_3 , LOW);
}
else if (load_choice <= 100){
    digitalWrite(Pinload_1 , HIGH);
    digitalWrite(Pinload_3 , LOW);
    digitalWrite(Pinload_2 , LOW);
}
}

```

```

// LCD UPDATING -----
if( (timer - LCD_timer) > 200 ){
  LCD_timer = timer;

  if(SCREEN == 0){

    LCD.setCursor(0, 0);
    LCD.print(BPM);
    LCD.print("BPM ");
    LCD.print(FREQ_HALL);
    LCD.print("HZ ");
    LCD.print(PEDALING);
    LCD.print(" ");
    LCD.setCursor(0, 1);
    LCD.print(load_choice);
    LCD.print("W ");
    LCD.print(output);
    LCD.print(" ");

    if ((output - FREQ_HALL) > 0)
      LCD.print("AUMENTAR");
    else if ((output - FREQ_HALL) < 0)
      LCD.print("DIMINUIR");
    else if ((output - FREQ_HALL) == 0)
      LCD.print("MANTER");

  }
  else if(SCREEN == 1){
    LCD.setCursor(0, 0);
    LCD.print("AVG: ");
    LCD.print(AVG_BPM);
    LCD.print("BPM ");
    LCD.print(" ");
  }
}

```



```

LCD.setCursor(0, 1);
LCD.print(AVG_LOAD);
LCD.print("W ");
LCD.print(output);
LCD.print(" ");
  if(BPM = frqcarrisk)
LCD.print("RISCO");
  else if(BPM = frqcarmax20)
LCD.print("MAX");
  else if(BPM = frqcarideal20)
LCD.print("IDEAL");
  else
LCD.print("MIN");

LCD.print(" ");
}
else if(SCREEN == 2){

LCD.setCursor(0, 0);
LCD.print(BPM);
LCD.print("/");
LCD.print(AVG_BPM);
LCD.print("BPM ");

LCD.setCursor(0, 1);
LCD.print(load_choice);
LCD.print("/");
LCD.print(AVG_LOAD);
LCD.print("W ");
  }
}
}
else if(MODE == MODE_DATA_ACQUISITION){

```

```

if(BUTTON2){
  BUTTON2 = 0;
  MODE = MODE_USER_CHOICE;
}
if(BUTTON1){
  BUTTON1 = 0;
  if(analogRead(LOAD_CHOICE_ANALOG_PIN) > 512){
    EEPROM.write(0, 0);
    Serial.println("ERASED!");
  }
  else{
    int addr = 0;
    byte value;
    Serial.println("");
    while( (addr < EEPROM.length()) && ( value = EEPROM.read(addr)) != 0 ){
      Serial.write(value);
      addr++;
    }
    Serial.println("");
    Serial.println("DONE");
  }
}

if( (timer - LCD_timer) > 200 ){
  LCD_timer = timer;

  LCD.setCursor(0, 0);
  LCD.print("DATA ACQUISITION          ");
  LCD.setCursor(0, 1);
  if(analogRead(LOAD_CHOICE_ANALOG_PIN) > 512){
    LCD.setCursor(0, 1);
    LCD.print("ERASE");
  }
  LCD.print("          ");
}

```

}

}

}