



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica e Informática

Controle de Sistemas de Refrigeração Visando Conforto Térmico e Eficiência Energética

Monaliza Correa Sobreira da Silva

Campina Grande, PB
Dezembro de 2018

Monaliza Correa Sobreira da Silva

Controle de Sistemas de Refrigeração Visando Conforto Térmico e Eficiência Energética

*Trabalho de Conclusão de Curso
apresentado à Coordenação do
Curso de Graduação em Engenharia
Elétrica da Universidade Federal de
Campina Grande, Campus Campina
Grande, como parte dos requisitos
necessários para a obtenção do grau
de Bacharel em Engenharia Elétrica.*

Área de Concentração: Eficiência Energética

Orientador: Edmar Candeia Gurjão

Campina Grande, PB
Dezembro de 2018

Monaliza Correa Sobreira da Silva

Controle de Sistemas de Refrigeração Visando Conforto Térmico e Eficiência Energética

*Trabalho de Conclusão de Curso
apresentado à Coordenação do
Curso de Graduação em Engenharia
Elétrica da Universidade Federal de
Campina Grande, Campus Campina
Grande, como parte dos requisitos
necessários para a obtenção do grau
de Bacharel em Engenharia Elétrica.*

Aprovado em ___/___/___

Luciana Ribeiro Veloso
Universidade Federal de Campina Grande
Avaliador

Edmar Candeia Gurjão
Universidade Federal de Campina Grande
Orientador

*A fé é o caminho que nos faz ver a vitória
ainda antes dela ter acontecido.*

Agradecimentos

Eu agradeço primeiramente a Deus, pelo dom da vida, por ser minha Força nos momentos em que eu já não sabia onde buscar e também por colocar verdadeiros anjos no meu caminho.

Agradeço a minha querida mãe, que nunca me deixou faltar nada, por sempre lutar para me proporcionar uma boa educação, me motivar e confiar em minhas escolhas acadêmicas. Agradeço aos meus avós, os quais me acolheram com todo amor durante esse período de curso e são figuras protagonistas na realização desse sonho. Agradeço à toda a minha família, que com todo carinho, me apoiaram e levantaram o ânimo em momentos difíceis.

Agradeço ao meu namorado Pedro, meu ponto de calma em meio às tempestades, me impulsionando nos desafios da graduação e sendo meu melhor amigo. Agradeço a todos os meus estimados amigos. Não se chega a lugar algum sozinho e tive a prova disso incontáveis vezes. Em especial, Marina Lua, Samuel Melo e Raphael Victor por serem, de diferentes formas, pessoas especiais, confiáveis e preocupadas com minha saúde mental.

Agradeço aos professores do Curso de Engenharia Elétrica da UFCG. Muitas vezes nos ensinando lições para a vida, além do conhecimento científico e prático inerente ao curso. Meu profundo agradecimento aos que fazem a Coordenação de Engenharia Elétrica. Pessoas profissionais, compreensivas e prontas a ajudar.

Agradeço ao prof. Edmar Gurjão pela orientação ao longo deste trabalho, por toda a paciência, incentivo e boas indicações. E, por fim, meus sinceros agradecimentos pelo auxílio dos colegas do LABMET.

*O sucesso é a soma de pequenos
esforços repetidos dia após dia
Robert Collier*

Resumo

O gerenciamento de energia elétrica pelo lado da demanda se apresenta como uma urgência devido ao aumento progressivo do consumo. Em residências, estima-se que o uso de condicionadores de ar, em épocas de calor, representa 60% do consumo de energia. Ajustar a temperatura destes aparelhos visando conforto térmico representa bem estar ao usuário e eficiência energética. Atendendo a essa necessidade, o controle de temperatura dos sistemas de refrigeração surge como solução. Neste trabalho foi aplicada a lógica fuzzy, a qual é caracterizada por ser uma lógica difusa, ou seja, seus valores são expressos linguisticamente, replicando o pensamento humano. Foram desenvolvidas regras para o controle da temperatura do ar-condicionado e realizados testes no LABMET na UFCG e constatou-se que o controle possui simplicidade na implementação, ótima precisão e rapidez na atuação.

Palavras chave: Temperatura, Lógica Fuzzy, Conforto Térmico.

Abstract

Management by the demand side appears as an urgency due to the progressive increase in electric energy consumption. In homes it is estimated that the use of air conditioners, in hot weather periods, represents 60% of energy consumption. Adjusting the temperature of these devices for thermal comfort represents user well-being and energy efficiency action. Thus, the temperature control of the refrigeration systems appears as a solution to this demand. In this work we apply the fuzzy logic, which is characterized by representing its values linguistically, replicating human thought. We developed the fuzzy rules and performed tests at LABMET at the UFCG and it was verified that the control has simplicity in the implementation, great precision and speed in the performance.

Keywords: Temperature, Fuzzy Logic, Thermal Confort.

Lista de Figuras

1	Faixas de pertinência da entrada Distância (Fonte: D. S. Kidri, 2012).	18
2	Curvas de representação (Fonte: Fachine, 2014).	19
3	Arquitetura de um controlador Fuzzy (Fonte: Marques, 2016).	19
4	Métodos de defuzzificação (Fonte: G. C. Santos, 2014).	20
5	Módulo sensor de temperatura e umidade - DHT11 (Fonte: Próprio Autor, 2018). .	21
6	Módulo receptor infravermelho (Fonte: Próprio Autor, 2018).	22
7	Módulo LED emissor infravermelho (Fonte: Próprio Autor, 2018).	23
8	Modelo de ar-condicionado do LABMET (Fonte: (MIDEA [®] , 2018).	23
9	IDE do Arduino (Fonte: Próprio Autor, 2018).	24
10	Erro apresentado ao compilar o código no Arduino Uno (Fonte: Próprio Autor, 2018).	25
11	Observação ao compilar o código no Arduino Mega (Fonte: Próprio Autor, 2018). .	25
12	Tela inicial do Matlab 2016a (Fonte: Próprio Autor, 2018).	26
13	Tela inicial do Fuzzy Logic Toolbox (Fonte: Próprio Autor, 2018).	27
14	Ajuste da variável de entrada IC (Fonte: Próprio Autor, 2018).	28
15	Ajuste da variável de entrada TempAtual (Fonte: Próprio Autor, 2018).	28
16	Ajuste da variável de saída TempIdeal (Fonte: Próprio Autor, 2018).	29
17	Tela de edição de regras no Fuzzy Logic Toolbox (Fonte: Próprio Autor, 2018). . .	30
18	Base de regras incorporada ao Fuzzy Logic Toolbox (Fonte: R. Alfaia, 2018). . . .	30
19	Visualização das regras (Fonte: Próprio Autor, 2018).	31
20	Gráfico de superfície (Fonte: Próprio Autor, 2018).	31
21	Esquema de ligações do projeto final (Fonte: Próprio Autor, 2018).	32
22	Arranjo para captação dos sinais do ar-condicionado (Fonte: Próprio Autor, 2018).	33
23	Projeto final do controlador (Fonte: Próprio Autor, 2018).	33
24	Projeto final do controlador (Fonte: Próprio Autor, 2018).	34
25	Processo desde a aquisição até a geração do resultado (Fonte: Próprio Autor, 2018).	34
26	Quadro de Temperaturas e Umidades (Fonte: INMET, 2018).	35
27	Visualizador de regras (Fonte: Próprio Autor, 2018).	36
28	Janela de comando do Matlab (Fonte: Próprio Autor, 2018).	36
29	Visor do ar-condicionado do LABMET (Fonte: Próprio Autor, 2018).	37

Lista de Tabelas

1	Categorias de conforto térmico (adaptado). (Fonte: ISO 7730, 2005).	15
2	Critérios para diferentes tipos de espaço (Fonte: ISO 7730, 2005).	16
3	Graus de cautela de acordo com o índice de calor (Fonte: Center, 1998).	17
4	Principais diferenças entre o Arduino Uno e o Mega 2560 (Fonte: Próprio Autor, 2018).	25
5	Prospecção do sistema de controle com dados INMET (Fonte: Próprio Autor, 2018).	38

Lista de Abreviaturas e Siglas

DR Draught Rate

IC Índice de Calor

ICSP In Circuit Serial Programming

INMET Instituto Nacional de Meteorologia

IR Infrared

ISO International Organization for Standardization

LABMET Laboratório de Metrologia

LED Light Emitting Diode

NBR Norma Brasileira

PMV Predicted Mean Vote

PPD Predicted Percentage of Dissatisfied

PROCEL Programa Nacional de Conservação de Energia Elétrica

PWM Pulse Width Modulation

SRAM Static Random Access Memory

UART Universal Asynchronous Receiver/Transmitter

UFMG Universidade Federal de Campina Grande

USB Universal Serial Bus

UTC Universal Time Coordinated

Sumário

Lista de Figuras	1
Lista de Tabelas	2
1 Introdução	13
2 Fundamentação Teórica	14
2.1 Conforto Térmico	14
2.2 ISO 7730/2005 - Ergonomia do ambiente térmico	15
2.3 Índice de Calor	17
2.4 Lógica Fuzzy	17
2.4.1 Fuzzificação	19
2.4.2 Inferência	20
2.4.3 Base de Conhecimento	20
2.4.4 Deffuzificação	20
3 Materiais e Métodos	20
3.1 Módulo Sensor de Temperatura e Umidade - DHT11	21
3.2 Módulo Receptor Infravermelho	21
3.3 Módulo LED Emissor Infravermelho	22
3.4 Ar-Condicionado	23
3.5 Arduino Mega 2560	24
3.6 Comunicação Serial - Arduino e Matlab	25
3.7 Matlab	26
3.7.1 Fuzzy Logic Toolbox	27
3.8 Estrutura Física do Projeto	32
4 Resultados e Discussões	34
5 Considerações Finais	39
6 Referências Bibliográficas	40
Anexo A	42
Anexo B	45
Anexo C	51

1 Introdução

Qualquer atividade na sociedade moderna só é possível com o uso extensivo de uma ou mais formas de energia. Dentre as diversas formas de energia nos interessam, em particular, aquelas que são processadas pela sociedade e colocadas à disposição dos consumidores onde e quando necessárias, tal como a eletricidade (INEE, 2017). Associado a essa disponibilidade, existem formas de gerenciar o uso da energia, uma das formas de realizar esse gerenciamento baseado na demanda do usuário é denominado de gerência pelo lado da demanda. O gerenciamento pelo lado da demanda se apresenta como uma urgência devido ao aumento progressivo do consumo de energia elétrica.

Nos países industrializados, práticas de gerenciamento pelo lado da demanda são mais comuns e estão incorporadas ao planejamento integrado. O Brasil se destaca pelo PROCEL (Programa Nacional de Conservação de Energia Elétrica), que visa o uso racional de energia elétrica e foi instituído em 30 de dezembro de 1985 pelos Ministérios de Minas e Energia e da Indústria e Comércio do Brasil (PROCEL, 2017). Este programa não atua diretamente no gerenciamento pelo lado da demanda, porém inclui ações dirigidas a conservação de energia elétrica, através da racionalização do uso e de incentivos ao desenvolvimento de produtos mais eficientes e de menor consumo.

O PROCEL, no entanto, direciona esforços aos segmentos comercial e industrial, além de iluminação pública e serviços de saneamento. Ou seja, não atua diretamente no segmento residencial. Em residências, existe a necessidade de gerenciamento voltado para a eficiência energética e que pode acarretar, conseqüentemente, na redução de custos.

Dentre as variáveis a serem trabalhadas para tornar a eficiência energética possível, destaca-se o controle da temperatura. A sensação de desconforto térmico, ainda mais presente nas estações quentes, implica no aumento do uso de condicionadores de ar, os quais representam 60% do consumo de energia em edifícios e residências (Chauduri, 2017). Um exemplo disso é a Universidade Federal de Campina Grande (UFCG), a qual conta com diversos prédios que somam 93 espaços somente em sua sede. Estima-se que o segundo maior gasto da instituição seja com energia elétrica e apesar de haverem diversos equipamentos em uso, é sabido que os condicionadores de ar representam um expressivo percentual desse consumo. Buscar medidas para diminuição desses valores de consumo é uma das práticas que estão sendo planejadas e estudadas, tanto no meio acadêmico, como no meio empresarial.

Como uma tentativa de reduzir o consumo de energia pelos ar-condicionados, o controle de temperatura em sistemas de refrigeração visa gerenciar o condicionador de ar para que mantenha o conforto térmico como prioridade e, conseqüentemente, haverá redução de custos em comparação ao uso normal do aparelho (R. Alfaia, 2018).

O objetivo desse trabalho é implementar um sistema inteligente que fará o controle do condi-

cionador de ar, usando as temperaturas internas e externas, o qual atuará no equipamento a fim de ajustar sua temperatura dentro de um padrão de conforto térmico. Para o desenvolvimento do trabalho foram obtidos dados de temperatura e umidade, tanto interna quanto externa ao ambiente, através de sensores. O algoritmo irá receber os dados e através da Lógica Fuzzy (D. Meana-Llorián, 2016) ajustará a temperatura dentro de um padrão de conforto térmico.

Foi usado como prova de conceito a sala de auditório do Laboratório de Metrologia de Campina Grande (LABMET), a qual detém de um condicionador de ar da marca MIDEA® e um visor embutido em sua estrutura para validar o controle da temperatura.

Este relatório é organizado em cinco capítulos. Este, o primeiro dos capítulos, contextualiza e cita os objetivos do trabalho. O segundo capítulo trata da fundamentação teórica, parte importante para a compreensão de conceitos necessários ao desenvolvimento do sistema de controle. O terceiro capítulo aborda os materiais e metodologia usados para elaboração do projeto. Ou seja, como é feita a aquisição das variáveis de entrada até a determinação da saída e atuação do controle. No quarto capítulo são expostos os resultados de funcionamento do controle em questão e prospecção de funcionamento do controle em um dia comum a fim de discutir o efeito positivo do sistema. No quinto e último capítulo são feitas considerações finais e sugestões de trabalhos futuros.

2 Fundamentação Teórica

Nesta seção serão apresentados os principais conceitos que fundamentaram este trabalho, que são: Conforto Térmico (2.1), Norma ISO 7730/2005 (2.2), Índice de Calor (2.3) e Lógica Fuzzy (2.4).

2.1 Conforto Térmico

Conforto térmico pode ser entendido como a condição ou sensação mental e física a qual o indivíduo se sente satisfeito no ambiente (A. P. Gagge, 1967). Dessa forma, num ambiente de conforto térmico espera-se que haja um maior aproveitamento das atividades ali executadas.

As condições de conforto térmico levam em conta diversas variáveis. Para avaliar tais condições, o indivíduo deve estar apropriadamente vestido e sem problemas de saúde ou de aclimação (adaptação fisiológica às mudanças ambientais). As condições ambientais capazes de proporcionar sensação de conforto térmico em habitantes de clima quente e úmido não são as mesmas que proporcionam sensação de conforto em habitantes de clima quente e seco e, muito menos, em habitantes de regiões de clima temperado ou frio.

No Brasil existem normas, como as NBR 16401-2, antiga NBR 6401, que trata de projetos de condicionares de ar visando conforto térmico, e a ISO 7730/2005, a qual estabelece diretrizes

para ergonomia do ambiente térmico, que definem, por exemplo, especificações de intervalos de temperatura do ambiente para o conforto térmico dos usuários. Portanto, esta última será a norma base para o desenvolvimento deste trabalho.

2.2 ISO 7730/2005 - Ergonomia do ambiente térmico

Devido às diferenças individuais, é impossível que um ambiente satisfaça termicamente a todas as pessoas. Sempre haverá uma porcentagem de pessoas que estarão insatisfeitas. É possível, no entanto, especificar ambientes que sejam aceitáveis termicamente, ou seja, satisfaçam a maioria de seus ocupantes.

Os propósitos dessa norma internacional são:

- Apresentar um método de cálculo da sensação térmica e o grau de desconforto das pessoas expostas a um ambiente térmico moderado;
- Especificar as condições de aceitabilidade térmica de um ambiente para conforto.

Tabela 1: Categorias de conforto térmico (adaptado). (Fonte: ISO 7730, 2005).

Categoria	Estado térmico do corpo		Desconforto local
	PPD %	PMV	DR %
A	<6	$-0.2 < PMV < +0.2$	<10
B	<10	$-0.5 < PMV < +0.5$	<20
C	<15	$-0.7 < PMV < +0.7$	<30

O **PPD** estabelece a quantidade estimada de pessoas insatisfeitas termicamente com o ambiente. Já o **PMV** é o voto de sensação de conforto térmico. Caracteriza-se por ser um índice que prevê o valor médio de um grande grupo de pessoas, segundo a escala de sensações de 7 pontos, onde +3 é muito quente e -3 é muito frio. Por último, o **DR** é a porcentagem de pessoas insatisfeitas devido à correntes de ar no local em análise (ISO, 2005).

Desse modo, após uma rápida verificação da Tabela 1, pode-se concluir que as variáveis apresentadas provém de parâmetros individuais e locais, ou seja, o desconforto térmico está associado ao lugar, mas também ao usuário. Destacam-se alguns parâmetros, tais como:

- Parâmetros Individuais
 - **Metabolismo**: O metabolismo corresponde à taxa de utilização de energia pelo corpo (ISO, 2005).

Controle de Sistemas de Refrigeração Visando Conforto Térmico e Eficiência Energética

- **Trabalho:** Tipicamente a avaliação de conforto verifica-se em casos em que o trabalho realizado é nulo ou corresponde apenas a dissipações por atrito, tais como, pessoas sentadas ou em andamento contínuo.
 - **Vestuário:** O tipo de vestimenta utilizada pelo indivíduo.
- **Parâmetros Ambientais**
 - **Temperatura do ar:** Corresponde a temperatura seca do ar, ou seja, sem considerar a umidade.
 - **Umidade do ar:** A umidade relativa pode variar de 0% (ausência de vapor de água no ar) a 100% (quantidade máxima de vapor de água que o ar pode dissolver, indicando que o ar está saturado). Em regiões onde a umidade relativa do ar se mantém muito baixa por longos períodos, as chuvas são escassas.
 - **Temperatura média radiante:** É a temperatura média à superfície dos elementos que envolvem um espaço. Influencia tanto o calor perdido através da radiação do corpo como a perda de calor por condução, quando o corpo está em contato com superfícies mais frias.

Conforme Tabela 2 são apresentadas temperaturas operativas de acordo com tipo de espaço e categoria. São derivados de hipóteses de atividades leves e vestimentas apropriadas, tanto no verão quanto no inverno. Os espaços não são restritos apenas aos apresentados, podendo expandir as análises para espaços similares.

Tabela 2: Critérios para diferentes tipos de espaço (Fonte: ISO 7730, 2005).

Tipo de espaço	Atividade W/m ²	Categoria	Temperatura Operativa °C	
			verão	inverno
Salas de aula Auditório Restaurante	70	A	24.5 ± 1	22 ± 1
		B	24.5 ± 1.5	22 ± 2
		C	24.5 ± 2.5	22 ± 3
Jardim de infância	81	A	23.5 ± 1	22 ± 1
		B	23.5 ± 2	22 ± 2.5
		C	24.5 ± 2.5	22 ± 3.5
Lojas	93	A	23 ± 1	19 ± 1.5
		B	23 ± 2	19 ± 3
		C	23 ± 3	19 ± 4

2.3 Índice de Calor

O índice de calor objetiva determinar o efeito da umidade relativa sobre a temperatura aparente do ar (Center, 1998). Em outras palavras, é uma medida que tem por objetivo inferir a intensidade do calor que uma pessoa sente. O resultado também é conhecido como "temperatura aparente" ou "sensação real" (G. B. Anderson, 2013). Esse índice tem como variáveis a temperatura e a umidade do ar. A fórmula a seguir reproduz o índice de calor e é uma adaptação da fórmula usada na tabela de índices de calor da Administração Oceânica e Atmosférica Nacional, USA (Center, 1998).

$$IC = \frac{5}{9}((c_1 + c_2T + c_3R + c_4TR + c_5T^2 + c_6R^2 + c_7T^2R + c_8TR^2 + c_9T^2R^2) - 32) \quad (1)$$

Onde:

IC = Índice de calor (em graus Celsius);

T = Temperatura (em graus Fahrenheit);

R = Umidade relativa (em porcentagem);

$$c_1 = -42.379$$

$$c_6 = -5.481717 \times 10^{-2}$$

$$c_2 = 2.04901523$$

$$c_7 = 1.22874 \times 10^{-3}$$

$$c_3 = 10.14333127$$

$$c_8 = 8.5282 \times 10^{-4}$$

$$c_4 = -0.22475541$$

$$c_9 = -1.99 \times 10^{-6}$$

$$c_5 = -6.83783 \times 10^{-3}$$

A interpretação dos resultados segue de acordo com a Tabela 3 :

Tabela 3: Graus de cautela de acordo com o índice de calor (Fonte: Center, 1998).

Índice de Calor (°C)	Grau de Cautela
27-32°C	Cuidado
33-39°C	Cuidado Extremo
40-51°C	Perigo
52-58°C	Perigo Extremo

2.4 Lógica Fuzzy

A lógica Fuzzy é utilizada para tratar problemas nos quais a imprecisão e a incerteza são variáveis que dificultam a implementação da maneira convencional (L. A. Prado, 2014). De modo figurativo, enquanto a lógica clássica enxerga somente o preto e branco, a logica fuzzy é capaz de, além do preto e branco, enxergar vários tons de cinza entre as duas cores.

Lofti A. Zadeh, em 1965, apresenta a teoria dos conjuntos fuzzy, a qual visa determinar um grau de pertinência que indica o quanto um determinado elemento pertence a um determinado conjunto,

ou seja, seu peso dentro daquele conjunto. Esse grau de pertinência é definido por meio de uma função característica real (função de pertinência) mapeada por

$$\mu_F \rightarrow [0, 1] \quad (2)$$

Associa a cada $x \in U$ um número real que está entre 0 e 1, ao invés de valores binários usuais.

Vários conjuntos fuzzy podem estar associados a um nome, chamado de variável linguística (Engelbrecht, 2014). Por exemplo, a distância de um determinado robô com relação a um obstáculo pode ser uma variável linguística assumindo os valores perto, médio e longe. Estes valores são descritos por intermédio de conjuntos fuzzy, representados por funções de pertinência (D. S. Kidri, 2012), conforme mostrado na Figura 1.

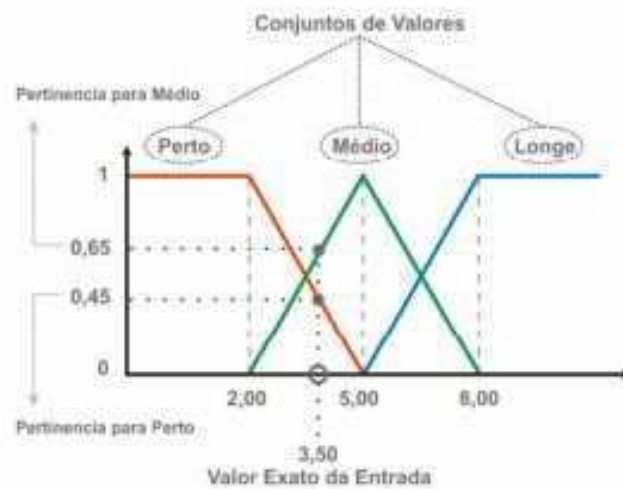


Figura 1: Faixas de pertinência da entrada Distância (Fonte: D. S. Kidri, 2012).

Nota-se que os conjuntos Perto e Longe são representados por um trapézio, enquanto o conjunto Médio é representado por um triângulo. No entanto, existem outras curvas de representação, como a gaussiana e a sino generalizada, conforme Figura 2.

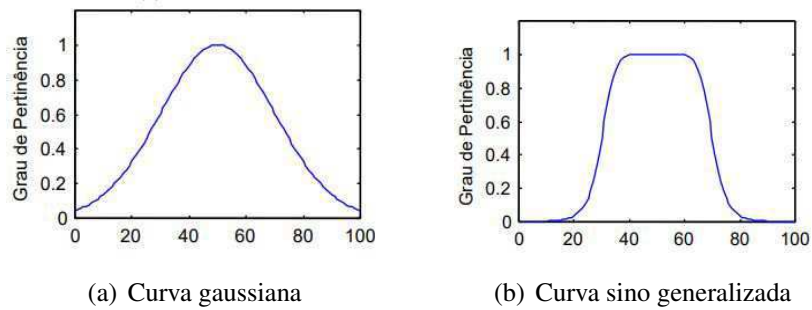


Figura 2: Curvas de representação (Fonte: Fechine, 2014).

Após a definição do grau de pertinência da variável de entrada, é hora de fazer o tratamento dessa informação. Os controladores convencionais usam modelagem matemática para fazer o controle dos processos. Já os controladores Fuzzy usam regras lógicas para inferir os resultados.

Os controladores Fuzzy realizam, basicamente, três etapas: **fuzzificação, inferência e defuzzificação**.

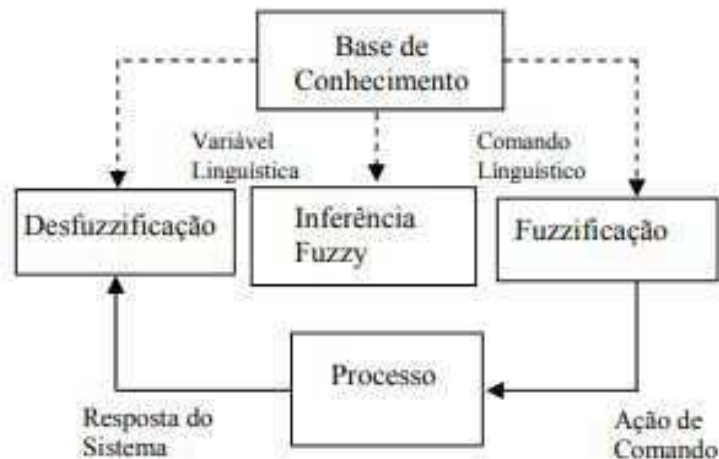


Figura 3: Arquitetura de um controlador Fuzzy (Fonte: Marques, 2016).

2.4.1 Fuzzificação

É o processo pelo qual a variável de entrada é associada a uma variável linguística. Na Figura 1 nota-se como o valor da entrada, já associado a variável linguística Distância, pertence a dois conjuntos, com graus de pertinência diferentes. O valor da entrada é 3,5 e pertence ao conjunto Perto, com grau de pertinência 0,45. Mas também pertence ao conjunto Médio com grau de pertinência 0,65 (D. S. Kidri, 2012).

2.4.2 Inferência

É caracterizado por associar as variáveis linguísticas de entrada e as variáveis linguísticas de saída. Isso ocorre pela identificação das regras que foram ativadas na base de regras.

2.4.3 Base de Conhecimento

É o "cérebro" do algoritmo. Representa o modelo do sistema a ser controlado, consistindo numa base de dados e uma base de regras Fuzzy linguísticas. A base de dados fornece definições numéricas e a base de regras caracteriza o controlador e sua estratégia usada, geralmente fornecida por pessoas especialistas no sistema.

Regras Fuzzy são definidas da seguinte forma: *Se Distancia é Perto então Velocidade é Lenta*, onde a variável linguística da entrada Distancia é Perto e a variável linguística da saída Velocidade é Lenta (D. S. Kidri, 2012).

2.4.4 Defuzzificação

Transforma as variáveis linguísticas de saída em valores numéricos a serem enviados aos atuadores. Existem vários métodos de defuzzificação na literatura, os mais empregados são o centro de gravidade (centroide) e a média dos máximos. Neste último, a saída é obtida tomando a média entre os dois elementos extremos no universo que correspondem aos maiores valores da função de pertinência de saída. Com o centro de gravidade, o valor numérico obtido representa o centro de gravidade da distribuição de possibilidade de saída do sistema fuzzy.

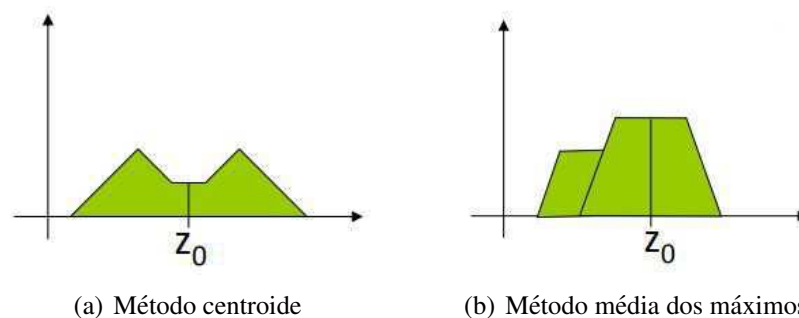


Figura 4: Métodos de defuzzificação (Fonte: G. C. Santos, 2014).

3 Materiais e Métodos

Nesta seção serão apresentados os materiais e procedimentos realizados para a concepção do controlador. O processo de teste e demonstração criado foi um sistema de controle de temperatura,

automatizado pelo Arduino, que é uma ferramenta prática para reger circuitos eletrônicos, sendo este então o meio físico pelo qual a automação acontece. O Matlab, por sua vez, processa essas informações de entrada, realiza a lógica fuzzy e envia o valor de saída para que o Arduino possa atuar.

3.1 Módulo Sensor de Temperatura e Umidade - DHT11

O módulo sensor de temperatura e umidade DHT11 permite fazer leituras de temperaturas de 0 a 50°C e umidade de 20 a 90%. Há um circuito interno que faz a leitura dos sensores e se comunica com o microcontrolador por um sinal serial de uma via. O módulo possui um tempo de resposta de aproximadamente dois segundos. No projeto foram usados dois módulos sensores. O primeiro objetiva captar a temperatura e umidade externa à sala e o segundo para captar a temperatura do interior da sala.

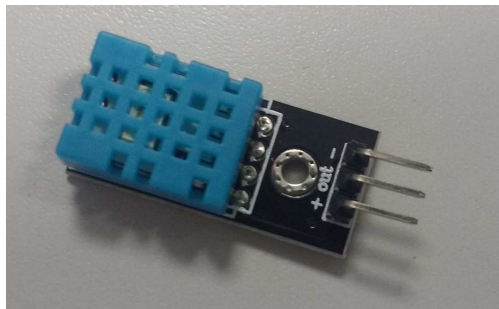


Figura 5: Módulo sensor de temperatura e umidade - DHT11 (Fonte: Próprio Autor, 2018).

3.2 Módulo Receptor Infravermelho

O módulo contém um receptor IR 1838 que funciona com tensões entre 2,7 e 5,5V comunicando-se com o microcontrolador por meio de um único pino de sinal. Uma imagem do módulo é apresentado na Figura 6.

- As especificações são:
 - Módulo Receptor IR
 - Alimentação: 2,7 à 5,5V DC
 - Frequência de operação: 38kHz
 - Ângulo de detecção: 90°
 - Temperatura de operação: -25 à 85°C
 - Dimensões: 7,4 x 6,4 x 5.1mm

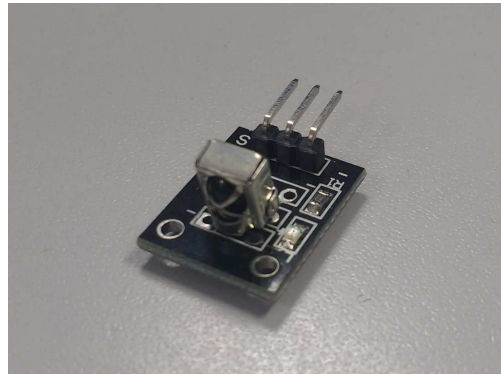


Figura 6: Módulo receptor infravermelho (Fonte: Próprio Autor, 2018).

Esse módulo servirá para captar os principais sinais de controle do condicionador de ar. Tais sinais são arrays de diferentes tamanhos e cada comando é um sinal distinto representado por uma matriz.

Foram captados os sinais mais importantes, como o de ligar, desligar, ajustar a temperatura em 20°, em 21° e assim por diante, em seguida foi necessário fazer um tratamento desses dados. Isso ocorre, pois diferente dos demais aparelhos que podem usar a biblioteca IRRemote do Arduino, o condicionador de ar não permite devido a grandeza do sinal captado. No Anexo A encontra-se o código usado para captar os sinais.

É válido salientar que os códigos captados para o modelo de ar-condicionado usado neste projeto poderão diferir dos demais aparelhos. Portanto, para cada condicionador de ar é preciso realizar o processo de captura de sinais.

3.3 Módulo LED Emissor Infravermelho

O Módulo LED Emissor Infravermelho (IR) KY-005 é um componente que emite um sinal IR que pode ser reconhecido por diversos receptores infravermelhos.

A faixa de luz infravermelha emitida pelo Módulo LED Emissor Infravermelho (IR) KY-005 não é perceptível ao olho humano, porém através de uma câmera de celular, por exemplo, é possível ver o sinal infravermelho sendo emitido pelo LED.

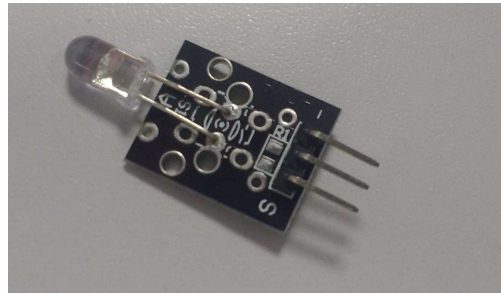


Figura 7: Módulo LED emissor infravermelho (Fonte: Próprio Autor, 2018).

- As especificações são:
 - Módulo LED Emissor IR
 - Alimentação: 5V DC
 - Comprimento da onda do LED IR: 940nm
 - Ângulo: 15° a 30°
 - Peso: 1g

Feito o tratamento dos sinais captados, resta testar sua eficácia emitindo-os através do LED. Aplicou-se o array no código que a própria biblioteca IRRemote do Arduino dispõe e, pela emissão dos sinais via LED Emissor, foi vista a atuação no condicionador de ar.

3.4 Ar-Condicionado

O controle foi testado no ar-condicionado da marca MIDEA situado na sala de auditório do LABMET. A escolha desse modelo em específico foi feita devido a necessidade de ter um display no próprio aparelho para certificação dos resultados. O modelo é o Elite 30k Frio.



Figura 8: Modelo de ar-condicionado do LABMET (Fonte: (MIDEA[®]), 2018).

- O modelo em questão possui as seguintes especificações (MIDEA[®], 2018):
 - Capacidade: 30000 BTU/h
 - Consumo: 2883W
 - Classificação Energética B

3.5 Arduino Mega 2560

O Arduino Mega 2560 é uma placa microcontroladora baseada no ATmega2560. Conta com 54 pinos de entrada / saída digital, dos quais 15 podem ser usados como saídas PWM. Também possui 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, um conector de energia, um conector ICSP, e um botão de reset. A placa Mega 2560 é compatível com a maioria dos estudos projetados para o UNO (ARDUINO[®], 2018).

Além disso, conta com uma Interface de Desenvolvimento - IDE multiplataforma, isto é, há a possibilidade de utilizar a IDE com sistemas operacionais distintos, como, Windows, Linux, Mac OS. Esta característica facilita o desenvolvimento dos algoritmos, podendo ser escritos, alterados e enviados ao Arduino de qualquer Sistema Operacional suportado.

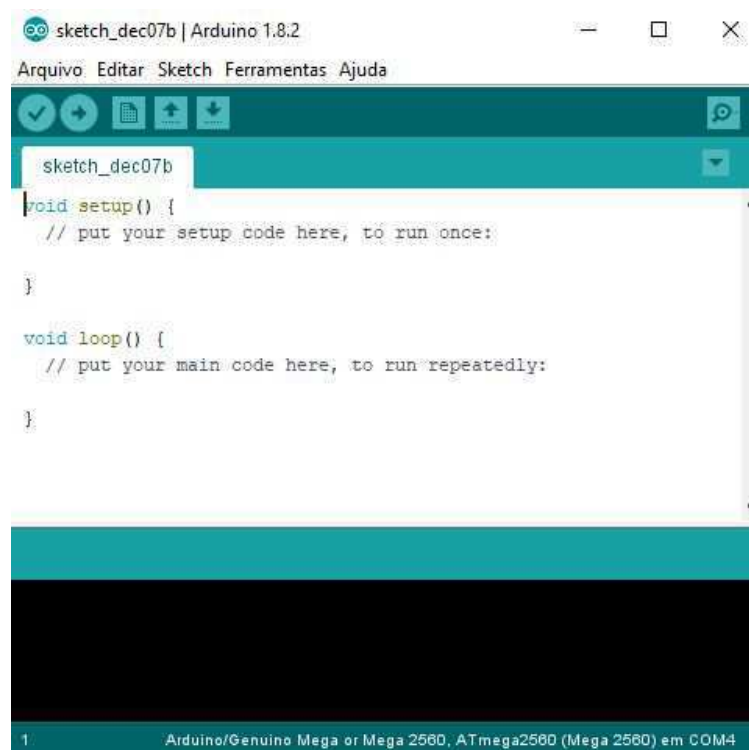


Figura 9: IDE do Arduino (Fonte: Próprio Autor, 2018).

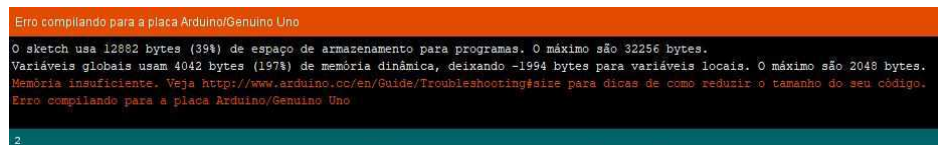
As placas microcontroladoras da marca são amplamente utilizadas no meio acadêmico e de fácil implementação, tendo um vasto acervo online de projetos disponíveis. Apesar do Arduino UNO

ser o campeão de uso no meio acadêmico, o trabalho em questão não pôde utilizá-lo devido a baixa capacidade de sua memória. De acordo com a Tabela 4, vê-se algumas das principais diferenças entre as placas.

Tabela 4: Principais diferenças entre o Arduino Uno e o Mega 2560 (Fonte: Próprio Autor, 2018).

	UNO	MEGA 2560
Microcontrolador	ATmega 328	ATmega 2560
Portas digitais	14	54
Portas PWM	6	15
Portas analógicas	6	16
Memória SRAM	2k bytes	8k bytes
Preço	\$ 22.00	\$ 38.50

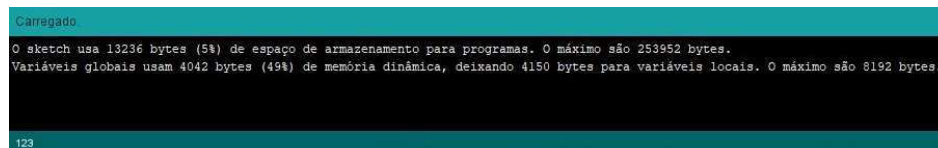
O microcontrolador ATmega328, do Arduino UNO, dispõe de uma memória SRAM de 2k bytes. No caso dos sinais do ar-condicionado serem arrays de várias posições do tipo unsigned int, o qual possuem um tamanho que varia de 2 a 4 bytes, a memória não comportou o programa e apresentou um erro de compilação. Da Figura 10 abaixo nota-se que as variáveis globais excederam 97% da memória da placa.



```
Erro compilando para a placa Arduino/Genuino Uno
O sketch usa 12882 bytes (39%) de espaço de armazenamento para programas. O máximo são 32256 bytes.
Variáveis globais usam 4042 bytes (197%) de memória dinâmica, deixando -1994 bytes para variáveis locais. O máximo são 2048 bytes.
Memória insuficiente. Veja http://www.arduino.cc/en/Guide/Troubleshooting#size para dicas de como reduzir o tamanho do seu código.
Erro compilando para a placa Arduino/Genuino Uno
```

Figura 10: Erro apresentado ao compilar o código no Arduino Uno (Fonte: Próprio Autor, 2018).

Já o ATmega 2560 serviu muito bem para a função requerida e satisfaz as necessidades de espaço para as variáveis globais, conforme Figura 11.



```
Carregado
O sketch usa 13236 bytes (5%) de espaço de armazenamento para programas. O máximo são 253952 bytes.
Variáveis globais usam 4042 bytes (49%) de memória dinâmica, deixando 4150 bytes para variáveis locais. O máximo são 8192 bytes.
```

Figura 11: Observação ao compilar o código no Arduino Mega (Fonte: Próprio Autor, 2018).

3.6 Comunicação Serial - Arduino e Matlab

A comunicação serial é o protocolo de baixo nível mais comum para comunicação entre dois ou mais dispositivos. Normalmente, um dispositivo é um computador, enquanto o outro dispositivo pode ser um modem, uma impressora, outro computador ou um instrumento científico, como um

osciloscópio ou um gerador de funções. Para muitos aplicativos de porta serial, é possível se comunicar com seu instrumento sem conhecimento detalhado de como a porta serial funciona (Souza, 2014).

A porta serial é inerente ao Arduino e o faz capaz de comunicar-se com um computador, através de uma entrada USB. O Matlab, por sua vez, pode receber ou enviar dados pela comunicação serial, e assim o controle é completo. A exploração desta comunicação foi feita neste projeto para enfatizar o que há de melhor em cada uma dessas ferramentas.

Antes de poder gravar ou ler dados, o objeto da porta serial e o instrumento devem ter configurações de comunicação idênticas.

3.7 Matlab

O *software* Matlab (acrônimo de MATrix LABoratory) é um software computacional conhecido mundialmente como uma excelente ferramenta para soluções de problemas matemáticos, científicos e tecnológicos. Foi aplicado no projeto em questão por ser muito fácil de usar, a linguagem é ideal para uso educacional e para desenvolver rapidamente protótipos de novos programas. Trata-se de um *software* interativo de alta performance que integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente intuitivo (MATHWORKS®, 2018).

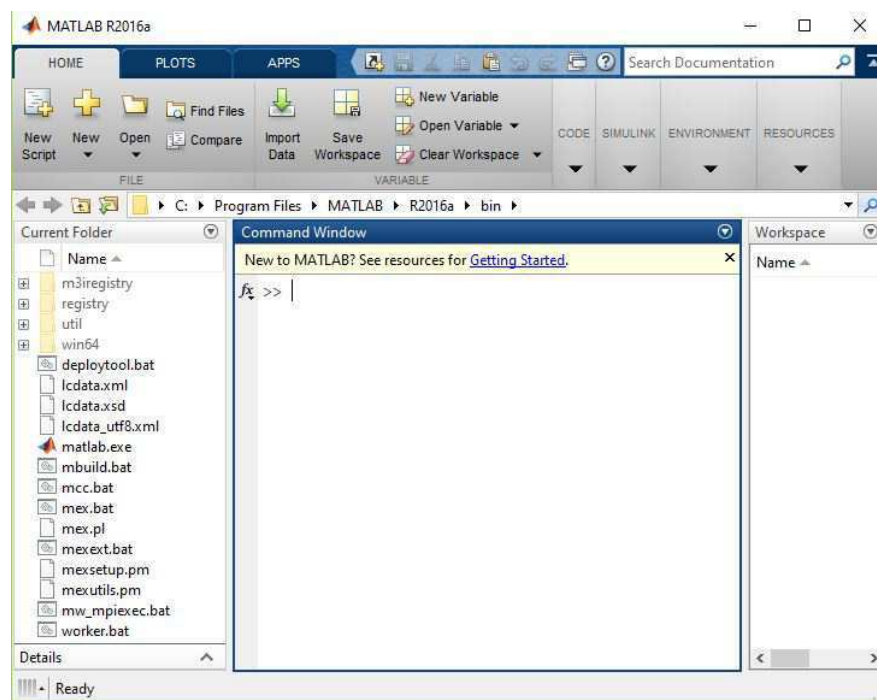


Figura 12: Tela inicial do Matlab 2016a (Fonte: Próprio Autor, 2018).

O programa dispõe de diversas extensões, chamadas toolboxes, que auxiliam no desenvolvimento específico de alguns problemas com interfaces próprias. Foi aplicada no desenvolvimento desse projeto a Fuzzy Logic Toolbox que permite que sejam modelados comportamentos complexos do sistema usando regras lógicas simples e, em seguida, implemente essas regras em um ambiente intuitivo.

Essa poderosa ferramenta foi essencial para o projeto, pois traz confiabilidade em seus resultados, além de recursos gráficos através de suas funções.

3.7.1 Fuzzy Logic Toolbox

A Fuzzy Logic Toolbox foi configurada para atender as necessidades do projeto e ser a ferramenta que desenvolve toda a lógica para que o Arduino estabeleça apenas comandos de atuação (MATHWORKS®, 2018). Inicialmente, foram adicionadas as variáveis. Neste caso, são duas variáveis de entrada, IC (índice de calor) e TempAtual (Temperatura Interna da sala), e uma variável de saída que é TempIdeal.

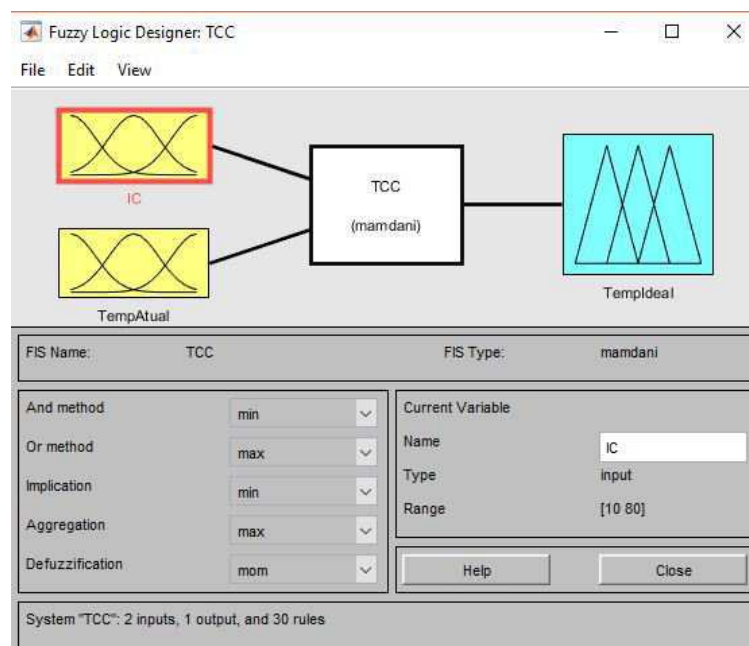


Figura 13: Tela inicial do Fuzzy Logic Toolbox (Fonte: Próprio Autor, 2018).

Em seguida, são ajustadas as variáveis. As funções de pertinência do Índice de Calor(IC) foram na Tabela 3 do Nacional Oceanic and Atmospheric Administration, no qual define faixas de temperatura representadas pelos seguintes intervalos: (i) frio quando se encontra abaixo de 21°C; (ii) confortável entre 10 e 32°C; (iii) desconfortável entre 22 e 40°C; (iv) quente entre 34 e 48°C; (v) super quente entre 42 e 58°C; (vi) risco a saúde quando se encontra superior a 50°C. A Figura 14 apresenta as funções de pertinência do IC.

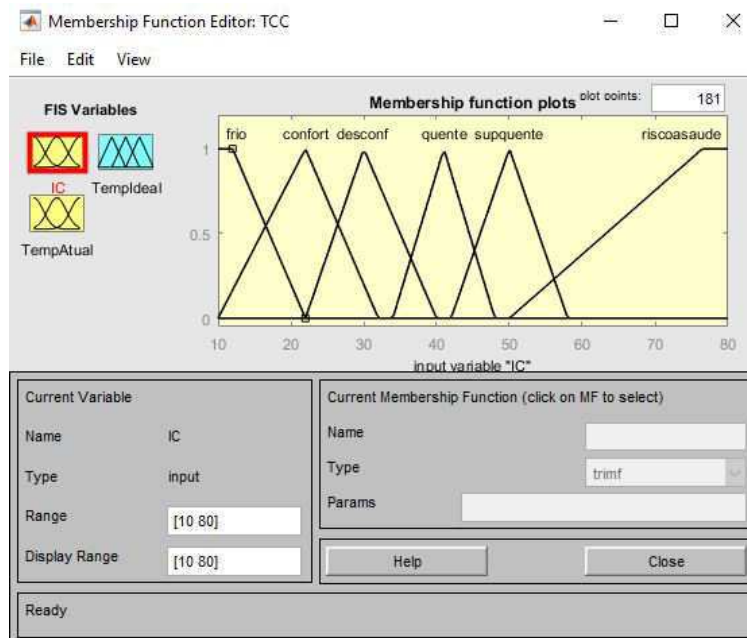


Figura 14: Ajuste da variável de entrada IC (Fonte: Próprio Autor, 2018).

A variável TempAtual é a segunda variável de entrada. Conta com cinco funções de pertinência, fundamentadas pelos conceitos de (HEATING, 2004), e são representadas pelos seguintes intervalos: (i) frio quando a temperatura interna está abaixo de 20°C; (ii) meio frio entre 15 e 25°C; (iii) confortável entre 20 e 30°C; (iv) meio quente entre 25 e 35°C; (v) quente quando encontra-se superior a 30°C;

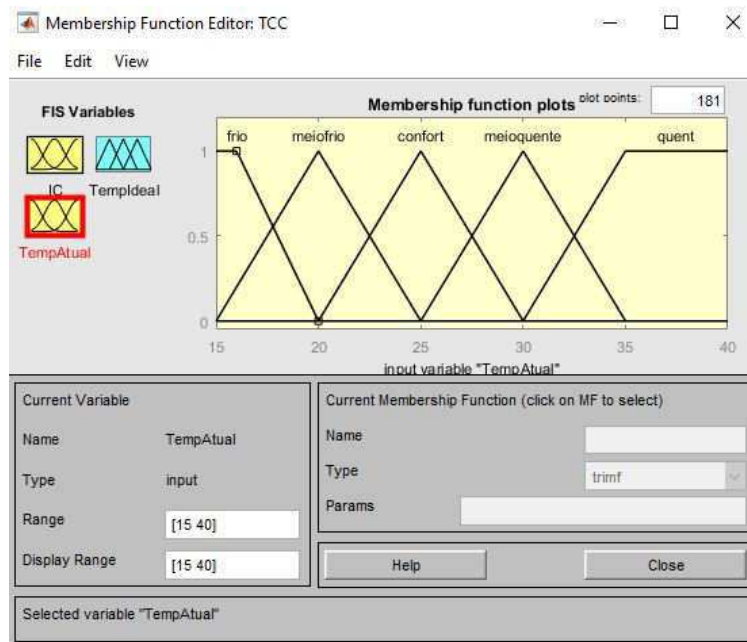


Figura 15: Ajuste da variável de entrada TempAtual (Fonte: Próprio Autor, 2018).

Por último, é preciso ajustar a variável de saída TempIdeal. Esta variável foi baseada nos conceitos da (ISO, 2005) apresentados na Tabela 2 e pode assumir os seguintes estados possíveis: (i) máxima quando assume valores menores que 19°C; (ii) fria entre 17 e 21°C; (iii) meio frio entre 19 e 23.5°C; (iv) confortável entre 21 e 26°C; (v) meio quente entre 24 e 28°C; (vi) desligado quando assume valores maiores que 26°C (ISO, 2005).

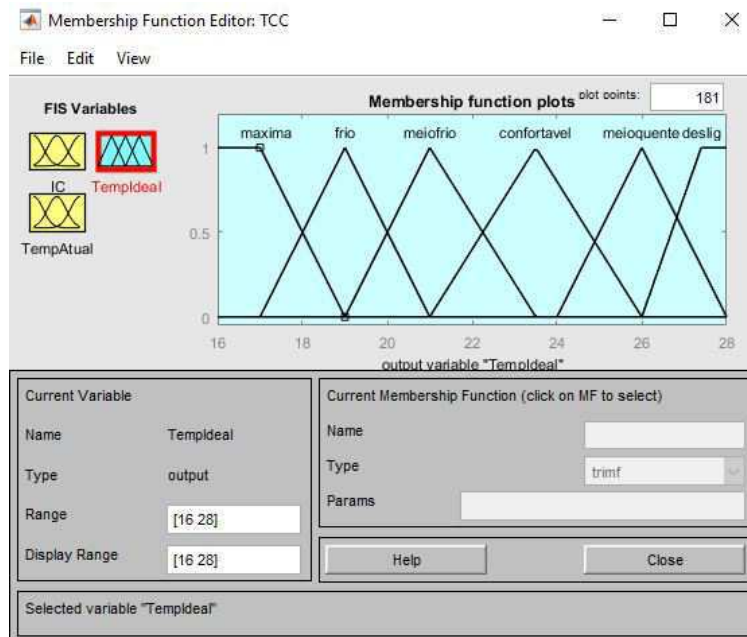


Figura 16: Ajuste da variável de saída TempIdeal (Fonte: Próprio Autor, 2018).

Controle de Sistemas de Refrigeração Visando Conforto Térmico e Eficiência Energética

Do ajuste das variáveis necessárias ao estudo, é hora de aplicar as regras onde a associação das variáveis de entrada dará origem a um resultado na saída.

Conforme Figura 17, o controle em questão possui 30 regras e isto o torna preciso, pois não deixa de associar nenhuma variável de entrada.

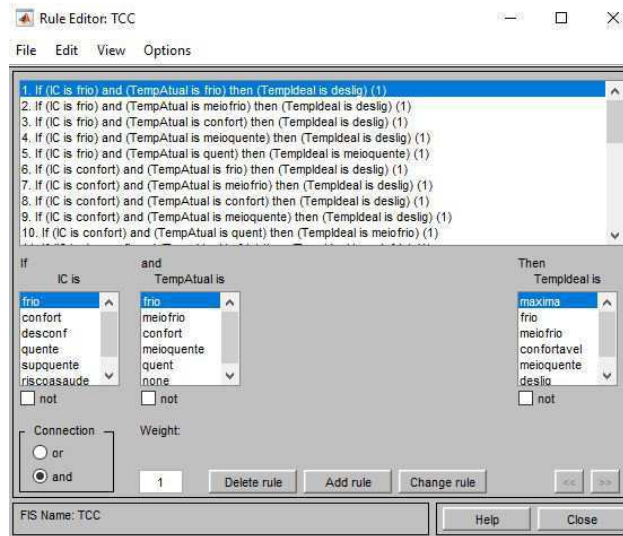


Figura 17: Tela de edição de regras no Fuzzy Logic Toolbox (Fonte: Próprio Autor, 2018).

Para melhor análise, segue a base de regras utilizada, conforme Figura 18.

		Temperatura Operante				
		Frio	Meio Frio	Confortável	Meio quente	Quente
Índice de Calor	Entradas					
	Frio	Desligado				M. Quente
	Confortável	Desligado				M. Frio
	Desconfortável	M. Frio	M. Frio	M. Frio	Frio	Frio
	Quente	Frio	Frio	M. Frio	Máximo	Máximo
	Super quente	Frio	Frio	Frio	Máximo	Máximo
Risco saúde	Frio	Máximo				

Figura 18: Base de regras incorporada ao Fuzzy Logic Toolbox (Fonte: R. Alfaia, 2018).

A Fuzzy Logic Toolbox dispõe de duas ótimas ferramentas de visualização: Rule Viewer e Surface Viewer. Na primeira é possível modificar os valores de entrada (Input), que estão entre colchetes, e observar o valor da saída. Torna-se de grande valia, pois demonstra a precisão da ferramenta, além de ser intuitivo. A segunda ferramenta dá uma visão geral de como são relacionadas as variáveis de entrada que originam a saída por meio de um gráfico de superfície. No Anexo C é fornecida a programação realizada no Matlab do presente projeto.

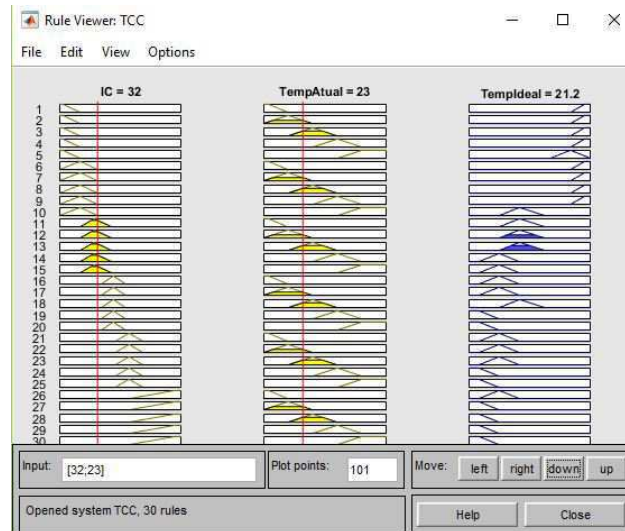


Figura 19: Visualização das regras (Fonte: Próprio Autor, 2018).

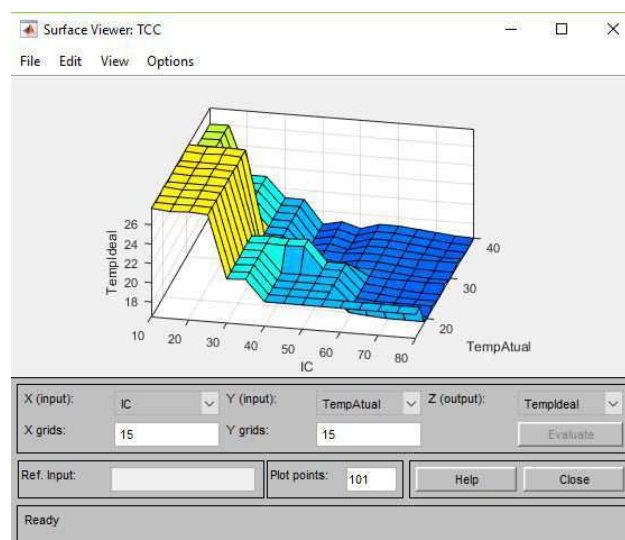


Figura 20: Gráfico de superfície (Fonte: Próprio Autor, 2018).

3.8 Estrutura Física do Projeto

Para que fosse possível executar o programa, foi necessário constituir um arranjo. Também chamado de *hardware*, o arranjo é a parte física do projeto e se refere aos dispositivos usados para realizar a atividade pretendida.

- Os materiais usados foram:
 - 1 Protoboard de 400 pontos
 - 2 Módulos sensores de temperatura e umidade DHT11
 - 1 Módulo Receptor infravermelho
 - 1 Módulo LED emissor infravermelho
 - 1 Cabo USB
 - Cabo vermelho 1.5mm
 - Cabo preto 1.5mm
 - Fita isolante
 - Jumpers

Um esquema de ligação foi idealizado para melhor entendimento das ligações, conforme abaixo.

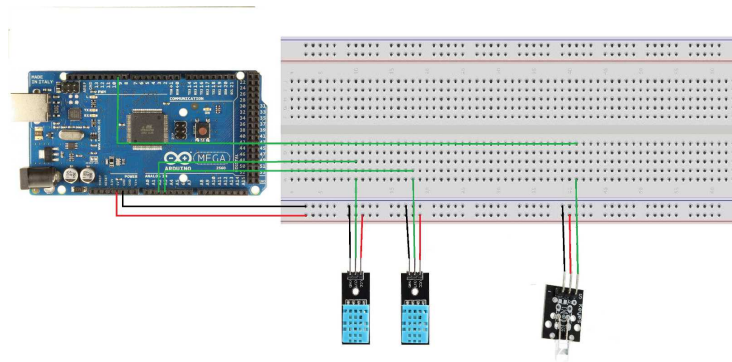


Figura 21: Esquema de ligações do projeto final (Fonte: Próprio Autor, 2018).

A seguir o *hardware* utilizado para a realização do projeto. Primeiramente, visualiza-se na Figura 22, o *hardware* utilizado para captar os sinais do ar-condicionado. Em seguida, o arranjo final do projeto para captação das temperaturas e umidades interna e externas ao auditório do LABMET. No Anexo B é fornecida a programação feita no Arduino para este projeto.

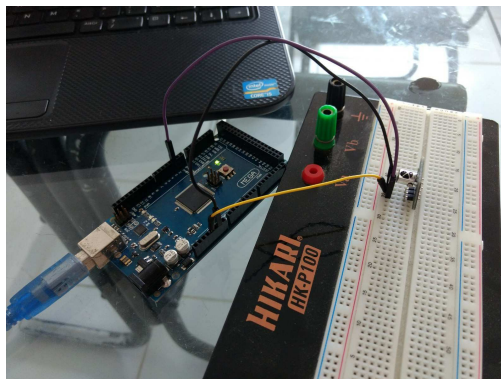


Figura 22: Arranjo para captação dos sinais do ar-condicionado (Fonte: Próprio Autor, 2018).

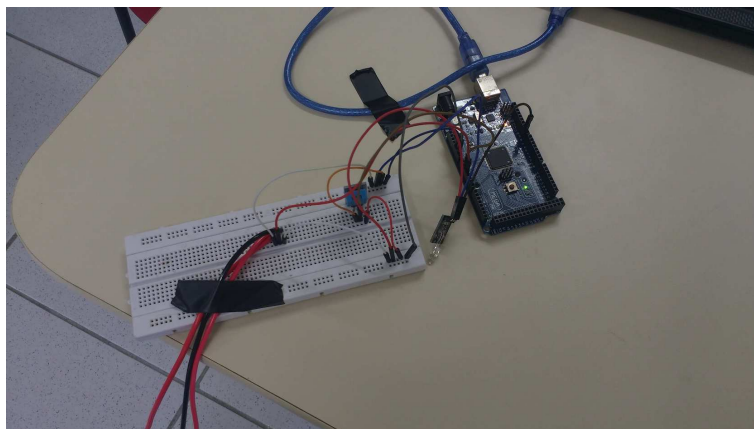
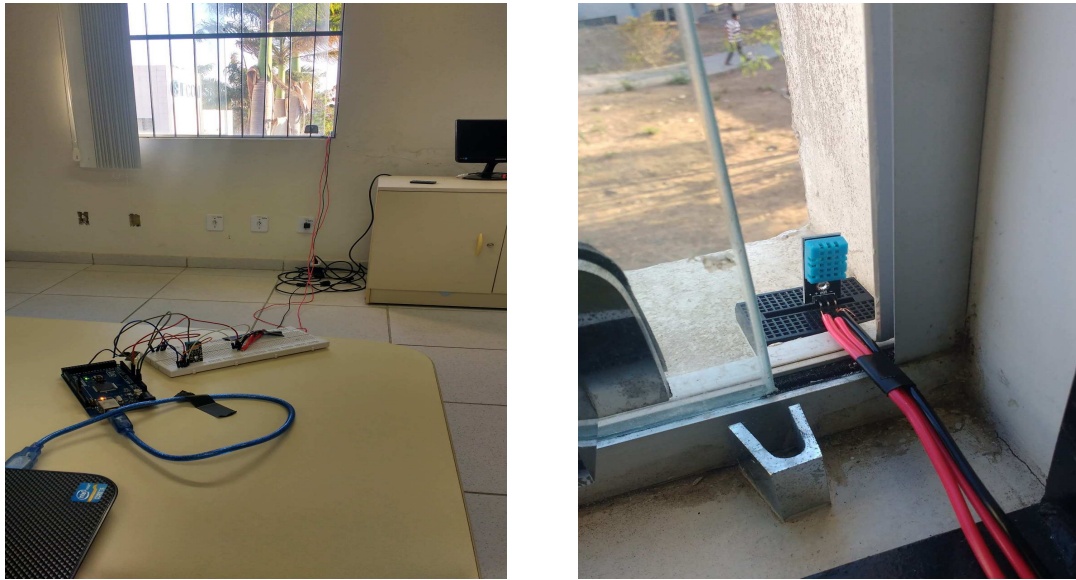


Figura 23: Projeto final do controlador (Fonte: Próprio Autor, 2018).



(a) Arranjo no interior da sala

(b) Sensor externo

Figura 24: Projeto final do controlador (Fonte: Próprio Autor, 2018).

4 Resultados e Discussões

O processo de evolução das variáveis foi idealizado no seguinte fluxograma:

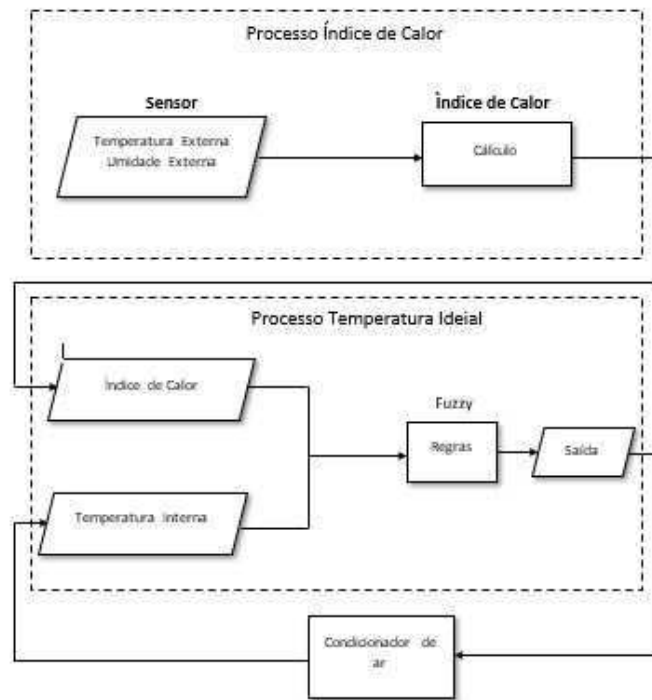


Figura 25: Processo desde a aquisição até a geração do resultado (Fonte: Próprio Autor, 2018).

Controle de Sistemas de Refrigeração Visando Conforto Térmico e Eficiência Energética

O teste foi realizado no auditório do LABMET dia 05 de dezembro de 2018 por volta das 14h. De acordo com dados do INMET (Instituto Nacional de Meteorologia) a temperatura naquele momento era de 31,5°C e umidade de 37%, conforme Figura 26. Note que o destaque está nas 17:00 UTC, equivalente a 15h horário de Brasília e 14h horário de Campina Grande.

O sensor de temperatura e umidade DHT11 posto do lado de fora do auditório marcava 30°C e umidade de 43%. Essa diferença pode ser justificada pela distância da Estação Meteorológica de Campina Grande para a UFCG.

Data	Hora	Temperatura (°C)			Umidade (%)		
		UTC	Inst.	Máx.	Mín.	Inst.	Máx.
05/12/2018	00	23.1	23.7	23.1	85	85	82
05/12/2018	01	22.9	23.2	22.9	87	87	85
05/12/2018	02	22.5	23.0	22.5	89	89	87
05/12/2018	03	22.5	22.6	22.4	90	90	89
05/12/2018	04	22.2	22.7	22.2	91	91	90
05/12/2018	05	22.4	22.4	22.2	91	91	91
05/12/2018	06	22.3	22.4	22.2	90	91	90
05/12/2018	07	22.3	22.4	22.2	91	91	90
05/12/2018	08	22.3	22.4	22.1	91	91	90
05/12/2018	09	22.7	22.7	22.2	90	91	90
05/12/2018	10	23.7	23.7	22.7	85	90	85
05/12/2018	11	25.6	25.6	23.7	74	85	74
05/12/2018	12	27.3	27.3	25.4	65	74	65
05/12/2018	13	29.6	29.9	27.1	53	65	53
05/12/2018	14	31.8	31.8	29.0	45	54	45
05/12/2018	15	32.1	32.2	29.8	41	48	40
05/12/2018	16	31.8	33.0	31.0	38	41	35
05/12/2018	17	31.5	32.3	31.1	37	40	36
05/12/2018	18	32.0	32.3	31.5	35	37	32
05/12/2018	19	28.5	32.2	28.4	60	61	34
05/12/2018	20	27.8	28.7	27.8	63	63	59
05/12/2018	21	26.1	27.8	26.0	72	72	63
05/12/2018	22	24.8	26.2	24.8	78	78	72
05/12/2018	23	23.9	24.8	23.9	82	82	78

Figura 26: Quadro de Temperaturas e Umidades (Fonte: INMET, 2018).

Tendo os valores de temperatura e umidade externas em 30°C e 43%, respectivamente, observou-se que o sensor de temperatura e umidade interno estava indicando uma temperatura de 23°C, uma vez que o ar-condicionado estava acionado há alguns minutos.

De modo que esses valores são passados via porta serial para o Matlab, eles são inseridos na lógica fuzzy e processados a fim de entregar uma saída para atuação do Arduino no ar-condicionado. De acordo com a Figura 27, nota-se que a saída coerente seria 21,2°C.

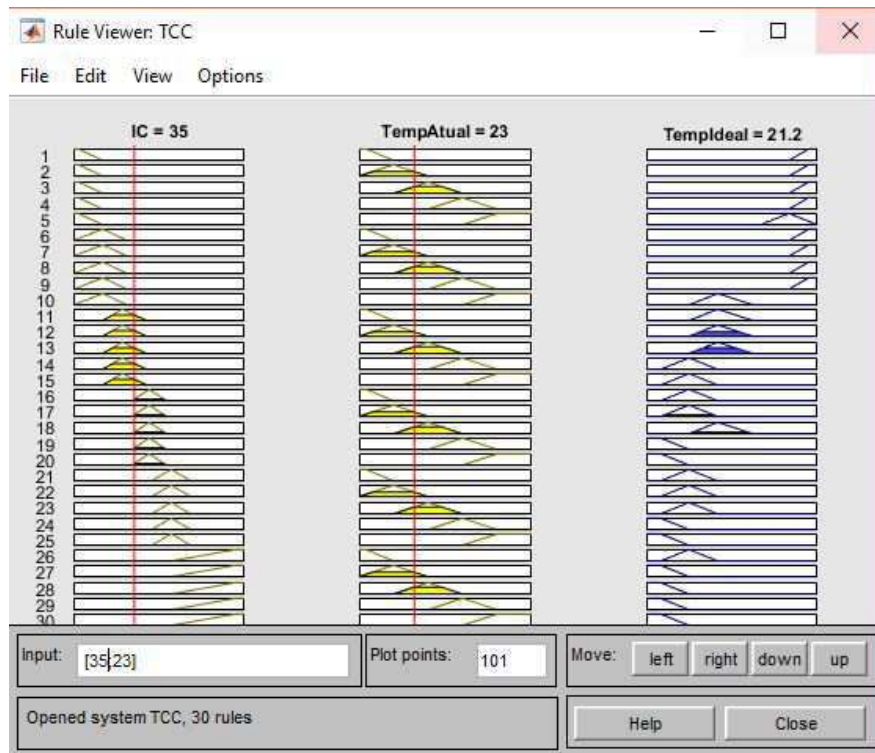


Figura 27: Visualizador de regras (Fonte: Próprio Autor, 2018).

Já a janela de comando do Matlab revela uma saída de 21°C, conforme Figura 28. Essa modificação é feita, pois a função `num2str` é utilizada e realiza esse arredondamento.



Figura 28: Janela de comando do Matlab (Fonte: Próprio Autor, 2018).

Após o envio desse resultado do Matlab para o Arduino, o mesmo atuou no ar-condicionado e ajustou sua temperatura, que estava em 23°C, para 21°C.

Nota-se, através dos testes, que o sistema de controle cumpriu seu papel de ajuste de temperatura baseado em conceitos de conforto térmico.

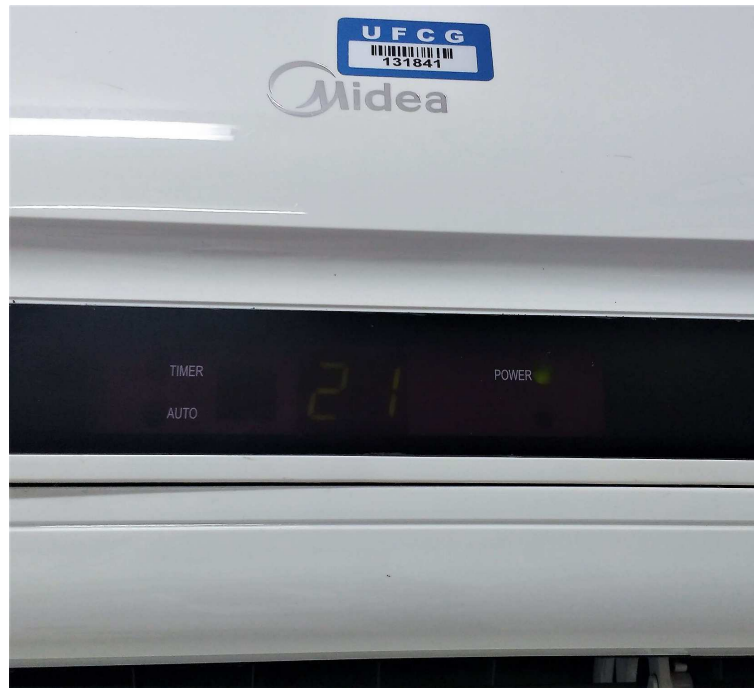


Figura 29: Visor do ar-condicionado do LABMET (Fonte: Próprio Autor, 2018).

Realizando uma análise do dia 05 de dezembro de 2018, é útil usar as informações oferecidas por (INMET, 2018) nos horários das 8h às 18h e prospectar um cenário de uso do sistema de controle.

No entanto, é recomendado fazer algumas considerações, tais como:

- O uso normal do ar-condicionado é fixo numa temperatura de 19°C;
- O sistema de controle é disparado a cada hora;
- Os dados dos sensores coincidem com os do INMET;
- No horário das 12h às 14h o ar-condicionado permanece ligado;

Desta feita, conforme Tabela 5, a saída para este dia, de acordo com o sistema de controle e no horário das 8 às 18h, seria de 21°C. Levando em consideração que, com o uso normal a temperatura ajustada no ar-condicionado seria menor, é possível inferir que haveria eficiência energética.

Controle de Sistemas de Refrigeração Visando Conforto Térmico e Eficiência Energética

Tabela 5: Prospecção do sistema de controle com dados INMET (Fonte: Próprio Autor, 2018).

Data	Hora		Temperatura (°C)	Umidade (%)	Saída de acordo com a Lógica Fuzzy
	UTC	CG	Instantânea	Instantânea	
05/12/2018	11	8	25.6	74	21.2
05/12/2018	12	9	27.3	65	21.2
05/12/2018	13	10	29.6	53	21.1
05/12/2018	14	11	31.8	45	21.2
05/12/2018	15	12	32.1	41	21.1
05/12/2018	16	13	31.8	38	21.2
05/12/2018	17	14	31.5	37	21.1
05/12/2018	18	15	32.0	35	21
05/12/2018	19	16	28.5	60	21.2
05/12/2018	20	17	27.8	63	21.1
05/12/2018	21	18	26.1	72	21.1

5 Considerações Finais

No presente trabalho foi elaborado um controle de temperatura do condicionador de ar do auditório do Laboratório de Metrologia, localizado na Universidade Federal de Campina Grande. Foi utilizada Lógica Fuzzy para processar as variáveis de entrada e, assim, visando o conforto térmico dos usuários, controlar a temperatura do ar-condicionado.

O controle de temperatura apresentou respostas satisfatórias e pode ser replicado para outros tipo de condicionadores de ar, bastando apenas captar os códigos correspondentes com receptor infravermelho.

Tendo em vista o caráter inovador da proposta, foram encontradas algumas dificuldades devido a ideia inicial ser um sistema embarcado. No entanto, a biblioteca fuzzy que o Arduino possui, a qual foi recentemente incorporada, não possui robustez para as 30 regras do projeto em questão. Desse modo, para que o projeto não perdesse confiabilidade, foi ponderada a utilização do Matlab, o que resultou em êxito na escolha.

Para um resultado mais consistente, seria de grande valia a análise do comportamento do controle ao longo de vários dias , e assim, computar em termos consumo o desempenho do condicionador de ar.

A aluna pôde desenvolver habilidades no âmbito de pesquisa, realizando uma vasta consulta na bibliografia, a fim de aprender sobre a lógica fuzzy. Com relação às simulações, foi de grande valia a utilização dos softwares para implementação da programação no Arduino e Matlab, a fim de consolidar o trabalho.

Para trabalhos futuros, pode-se apontar: elaboração de um controle de temperatura embarcado visando conforto térmico. Ademais, há espaço para realizar a implementação em uma placa controladora diferente ou em um ambiente que tenha um tratamento especial com a temperatura, como hospitais.

6 Referências Bibliográficas

- J. D. Hardy A. P. Gagge, J. A. J. Stolwijk. Comfort and thermal sensations and associated physiological responses at various ambient temperatures. 1967.
- ARDUINO®. Arduino - Comunicação Serial. <https://store.arduino.cc/usa/arduino-mega-2560-rev3>, 2018. Acesso: 3 de Dezembro de 2018.
- Damage Assessment Center. Noaa (national oceanic and atmospheric administration). us department of commerce. 1998.
- T. Chauduri. Machine learning based prediction of thermal comfort in buildings of equatorial singapore. In *IEEE International Conference on. IEEE*, pages 72–77, 2017.
- B. C. P. G-Bustelo J. M. Cueva Lovelle N. Garcia-Fernandez D. Meana-Llorián, C. González García. Iofclime: The fuzzy logic and the internet of things to control indoor temperature regarding the outdoor ambient conditions. *Futur. Gener. Comput. Syst*, 2016.
- M. V. S. Lemos R. A. L Rabêlo D. S. Kidri, A. J. O. Alves. Desenvolvimento de uma biblioteca fuzzy para o controle autônomo de um robô mível em ambiente desconhecido. *SBA Controle & Automação*, 2012.
- A. P. Engelbrecht. Aplicação de lógica fuzzy e arduino em controle de processos. *Trabalho de Pós Graduação*, 2014.
- J. M. Fachine. Representação do conhecimento (lógica fuzzy). 2014.
- R. D. Peng G. B. Anderson, M. L. Bell. Methods to calculate the heat index as an exposure metric in environmental health research. *environmental health perspectives*. 2013.
- F. M. Ribeiro J F. Araújo V. L. D. Mattos G. C. Santos, P. S. Thomaz. Influência do método de defuzzificação em mensurações com controladores fuzzy. In *XVII Simpósio de Pesquisa Operacional e Logística da Marinha*, 2014.
- AMERICAN SOCIETY OF HEATING. Thermal environmental conditions for human occupancy. 2004.
- INEE. Instituto Nacional de Eficiência Energética. http://www.inee.org.br/eficiencia_o_que_eh.asp?Cat=eficiencia, 2017. Acesso: 4 de Novembro de 2018.
- INMET. Instituto nacional de meteorologia. Campina Grande, 2018.

- ISO. Ergonomics of the thermal environment-analytical determination and interpretation of thermal comfort using calculation of the pmv and ppd indices and local thermal comfort criteria. 2005.
- Y. M. C. Marsselli L. A. Prado. Aplicação de lógica fuzzy e arduino em controle de processos. *Trabalho de Pós Graduação*, 2014.
- J. E. S. Marques. Aplicação da lógica fuzzy no controle da velocidade de um protótipo móvel. *CD-Revista Eletrônica da Fainor*, v.9, pages 15–25, 2016.
- MATHWORKS®. MathWorks.com. https://www.mathworks.com/products/fuzzy-logic.html?s_tid=srchtitle, 2018. Acesso: 3 de Dezembro de 2018.
- MIDEA®. <http://www.mideadobrasil.com.br/pt/produtos/interna/144/ar-condicionado-split-elite-30k-frio-30.000-btu-h>, 2018. Acesso: 3 de Dezembro de 2018.
- PROCEL. Programa Nacional de Conservação de Energia Elétrica. <http://www.procelinfo.com.br>, 2017. Acesso: 4 de Novembro de 2018.
- A. Souto M. Silva C. R. Francês R. Alfaia, E. Oliveira. Inclusão do módulo de gerenciamento inteligente na arquitetura smartcom, visando o gerenciamento eficiente do consumo de energia em sistemas de refrigeração em smart home, baseado em iot. In *Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, Campina Grande, 2018.
- F. Souza. <https://www.embarcados.com.br/arduino-comunicacao-serial/>, 2014. Acesso: 3 de Dezembro de 2018.

Anexo A

```
/*
Author: AnalysIR
Revision: 1.0

http://www.analysir.com/blog/2014/03/19/air-conditions-problems-recording-long-infrared-remote-control-signals-arduino/
```

```
Connections:
IR Receiver      Arduino MEGA
V+                -> +5v
GND               -> GND
Signal Out       -> Digital Pin 9
*/
```

```
#define LEDPIN 13
#define maxLen 800
```

```
volatile unsigned int irBuffer[maxLen]; //stores timings - volatile because
changed by ISR
volatile unsigned int x = 0; //Pointer thru irBuffer - volatile because changed
by ISR
```

```
void setup() {
  Serial.begin(115200); //change BAUD rate as required
  attachInterrupt(0, rxIR_Interrupt_Handler, CHANGE); //set up ISR for receiving
  IR signal
}
```

```
void loop() {

  Serial.println(F("Press the button on the remote now - once only"));
  delay(5000); // pausa de 5 segundos
  if (x) { //Se o sinal foi capturado
    digitalWrite(LEDPIN, HIGH); //Indica2o visual que recebeu o sinal
    Serial.println();
    Serial.print(F("Raw: "));
    Serial.print((x - 1));
    Serial.print(F(" "));
    detachInterrupt(0);
    for (int i = 1; i < x; i++) {
      if (!(i & 0x1)) Serial.print(F("-"));
      Serial.print(irBuffer[i] - irBuffer[i - 1]);
      Serial.print(F(", "));
    }
    x = 0;
    Serial.println();
    Serial.println();
    digitalWrite(LEDPIN, LOW); //Termina a indica2o visual
```

```
attachInterrupt(0, rxIR_Interrupt_Handler, CHANGE); //reabilita ISR para
receber novo sinal IR
}

}

void rxIR_Interrupt_Handler() {
  if (x > maxLen) return;
  irBuffer[x++] = micros();
}
```

Anexo B

```

/* Comunicaç2o Serial: L@gica Fuzzy com MATLAB e Ardui no
   Mbnaliza Correa Sobreira da Silva (Dez. 2018) - TCC UFCG */
#include "DHT.h" // Inclus2o da biblioteca do sensor de temperatura e umidade
DHT11
#define DHTPIN A3 // Conex2o pi no anal @gi co A3
#define DHTP A2 // Conex2o pi no anal @gi co A2
#define DHTTYPE DHT11 // DHT 11
#include <IRremote.h> // Inclus2o da bibliot eca para LED emissor infravermelho
IRsend irsend;
DHT dht( DHTPIN, DHTTYPE);
DHT dht2( DHTP, DHTTYPE);

float sai da;
int khz = 38;

const int ledpin = 13;

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  dht.begin();
  dht2.begin();
}

void enviaDadosMatlab() {
  // Leituras dos sensores e c@culo da vari @vel IC
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float h2 = dht2.readHumidity();
  float t2 = dht2.readTemperature();
  float t_f = (t * 9) / 5 + 32;
  double a = -42.379 + 2.04901523 * t_f + 10.14333127 * h - 0.22475541 * t_f * h
- 0.00683783 * pow(t_f, 2) - 0.05481717 * pow(h, 2) + 0.00122874 * pow(t_f, 2) *
h + 0.000885282 * t_f * pow(h, 2) - 0.00000199 * pow(t_f, 2) * pow(h, 2);
  double IC = 0.5556 * ( a - 32);
  double soma = t2 + IC;

  // Envio das vari @veis para a porta serial
  Serial.println(88);
  Serial.println(t);
  Serial.println(t2);
  Serial.println(IC);
  Serial.println(soma);
}

void RecebeDadosMatlab() {
//Recebe a sa da atravs da porta serial
if (Serial.available() > 0) {
  String saidastr = Serial.readString();
}
}

```

```

    sai da = sai da str.toFl oat();
}

}

void loop() {

    //Executa as funções
    enviaDadosMatlab();
    RecebeDadosMatlab();

    // Condições para ajustar a temperatura do ar condicionado de acordo com a
    saºda recebi da pelo Matlab.

    if ( sai da < 18) { //Ajustar em 17ºC
        unsigned int irSignalz[] = {4440, 4380, 568, 1648, 568, 532, 540, 1652, 544,
1648, 544, 556, 540, 560, 540, 1648, 544, 556, 544, 556, 544, 1652, 544, 556,
540, 560, 540, 1624, 592, 1624, 540, 556, 544, 1632, 564, 560, 540, 556, 544,
556, 544, 1648, 564, 1624, 544, 1648, 540, 1648, 544, 1652, 540, 1632, 564, 1648,
540, 1624, 568, 556, 544, 556, 540, 560, 540, 556, 544, 560, 540, 560, 568, 532,
540, 560, 540, 556, 544, 556, 544, 1652, 540, 556, 544, 560, 540, 1656, 540,
1624, 568, 1624, 568, 1648, 540, 1648, 544, 556, 540, 1632, 564, 1624, 564, 5252,
4440, 4384, 564, 1628, 564, 556, 544, 1652, 540, 1648, 544, 556, 544, 556, 540,
1648, 544, 560, 540, 560, 544, 1624, 568, 556, 544, 556, 544, 1648, 540, 1648,
544, 556, 544, 1628, 568, 556, 544, 556, 544, 556, 540, 1648, 544, 1648, 540,
1624, 568, 1648, 540, 1628, 564, 1628, 568, 1648, 544, 1624, 564, 556, 544, 556,
544, 556, 544, 556, 540, 560, 540, 560, 540, 560, 540, 560, 540, 560, 540, 556, 544, 556,
544, 1652, 540, 556, 544, 560, 540, 1632, 564, 1648, 568, 1624, 540, 1648, 544,
1624, 588, 536, 540, 1652, 544, 1648, 516};

        irsend.sendRaw(irSignalz, sizeof(irSignalz) / sizeof(irSignalz[0]), khz);
    }

    else if ( sai da < 19) { //Ajustar em 18ºC
        unsigned int irSignalz[] = {4436, 4396, 568, 1628, 564, 556, 544, 1652, 540,
1624, 568, 556, 540, 560, 540, 1624, 568, 560, 540, 560, 540, 1656, 540, 556,
544, 556, 544, 1648, 540, 1648, 544, 556, 544, 1628, 568, 556, 544, 556, 544,
556, 540, 1628, 564, 1648, 544, 1648, 540, 1648, 568, 1624, 544, 1656, 540, 1624,
564, 1652, 540, 556, 568, 532, 544, 556, 540, 560, 540, 560, 544, 556, 544, 556,
544, 556, 544, 1628, 564, 556, 544, 1628, 564, 560, 540, 560, 540, 1656, 540,
1628, 564, 1648, 540, 560, 540, 1652, 544, 556, 544, 1652, 540, 1624, 564, 5240,
4440, 4384, 568, 1648, 540, 560, 540, 1632, 564, 1624, 568, 556, 540, 556, 544,
1624, 564, 560, 544, 556, 544, 1628, 564, 560, 540, 556, 544, 1648, 540, 1628,
564, 556, 544, 1656, 540, 560, 540, 556, 544, 556, 544, 1648, 540, 1648, 568,
1624, 540, 1624, 568, 1648, 544, 1652, 544, 1648, 540, 1652, 540, 560, 540, 556,
544, 556, 544, 556, 540, 560, 540, 560, 544, 556, 544, 556, 540, 1652, 544, 556,
544, 1652, 540, 556, 544, 560, 540, 1628, 568, 1648, 544, 1624, 564, 556, 544,
1652, 544, 556, 540, 1632, 564, 1652, 540};

        irsend.sendRaw(irSignalz, sizeof(irSignalz) / sizeof(irSignalz[0]), khz);
    }
}

```

```
}
```

```
else if ( saida < 20) { //Ajustar em 19°C
```

```
    unsigned int irSignalz[] = {4440, 4392, 568, 1652, 540, 532, 568, 1652, 540,  
1652, 540, 532, 568, 528, 568, 1652, 540, 532, 568, 532, 568, 1652, 544, 528,  
568, 532, 568, 1648, 540, 1652, 540, 532, 568, 1652, 544, 536, 564, 532, 568,  
532, 564, 1652, 540, 1648, 540, 1652, 540, 1648, 540, 1652, 540, 1656, 540, 1624,  
568, 1648, 540, 532, 568, 556, 540, 536, 564, 532, 568, 536, 564, 536, 564, 536,  
564, 1624, 568, 1648, 540, 536, 564, 1652, 540, 560, 540, 536, 564, 1656, 540,  
1648, 544, 556, 540, 532, 568, 1624, 568, 532, 568, 1652, 540, 1648, 540, 5248,  
4440, 4384, 568, 1648, 540, 532, 568, 1652, 540, 1628, 560, 536, 564, 532, 568,  
1648, 540, 536, 568, 532, 568, 1652, 540, 532, 568, 556, 540, 1628, 564, 1648,  
544, 528, 568, 1656, 540, 536, 568, 528, 568, 536, 564, 1648, 544, 1624, 564,  
1624, 564, 1648, 544, 1648, 544, 1656, 540, 1648, 540, 1648, 544, 532, 564, 560,  
540, 532, 568, 432, 664, 536, 568, 532, 568, 532, 568, 1648, 540, 1648, 540, 532,  
572, 1652, 540, 556, 544, 536, 564, 1652, 544, 1648, 540, 532, 568, 532, 568,  
1648, 540, 532, 568, 1628, 564, 1652, 540};
```

```
    irsend.sendRaw(irSignalz, sizeof(irSignalz) / sizeof(irSignalz[0]), khz);
```

```
}
```

```
else if ( saida < 21) { //Ajustar em 20°C
```

```
    unsigned int irSignalz[] = {4436, 4392, 572, 1624, 560, 560, 540, 1652, 544,  
1648, 564, 532, 544, 556, 540, 1648, 540, 560, 544, 560, 540, 1624, 568, 556,  
544, 556, 544, 1644, 548, 1620, 564, 556, 540, 1632, 564, 560, 540, 560, 540,  
556, 544, 1624, 564, 1648, 540, 1648, 544, 1648, 564, 1624, 544, 1652, 544, 1624,  
564, 1648, 540, 556, 544, 556, 544, 556, 540, 556, 568, 532, 544, 560, 540, 556,  
544, 1648, 540, 556, 540, 560, 540, 1624, 568, 556, 540, 560, 540, 1632, 564,  
1624, 564, 560, 540, 1628, 564, 1648, 544, 556, 540, 1652, 544, 1648, 540, 5248,  
4436, 4384, 568, 1648, 540, 556, 544, 1628, 564, 1648, 544, 556, 540, 556, 544,  
1648, 540, 560, 540, 560, 540, 1652, 544, 556, 540, 556, 544, 1648, 544, 1620,  
568, 556, 544, 1628, 568, 556, 544, 556, 544, 556, 540, 1624, 568, 1624, 564,  
1648, 540, 1652, 540, 1648, 544, 1652, 544, 1624, 564, 1624, 568, 556, 540, 556,  
544, 556, 544, 556, 540, 560, 540, 560, 544, 556, 548, 1616, 568, 556, 544, 556,  
544, 1644, 544, 556, 544, 556, 544, 1628, 564, 1652, 540, 556, 544, 1628, 564,  
1624, 564, 560, 540, 1652, 544, 1652, 536};
```

```
    irsend.sendRaw(irSignalz, sizeof(irSignalz) / sizeof(irSignalz[0]), khz);
```

```
}
```

```
if ( saida < 22) { //Ajustar em 21°C
```

```
    unsigned int irSignalz[] = {4440, 4380, 568, 1648, 540, 560, 540, 1652, 540,  
1648, 544, 532, 564, 532, 568, 1648, 540, 560, 544, 532, 568, 1652, 540, 560,  
540, 536, 564, 1648, 540, 1648, 540, 560, 540, 1652, 544, 560, 540, 556, 544,  
556, 540, 1648, 544, 1644, 544, 1652, 540, 1644, 544, 1648, 544, 1652, 544, 1644,  
544, 1648, 540, 560, 540, 536, 564, 556, 540, 560, 540, 560, 540, 536, 568, 1648,  
544, 1648, 540, 560, 540, 556, 544, 1644, 544, 556, 540, 560, 544, 1628, 568,  
556, 540, 560, 540, 1648, 540, 1648, 540, 560, 540, 1652, 544, 1644, 544, 5248,  
4436, 4384, 568, 1644, 544, 556, 540, 1628, 568, 1644, 544, 556, 544, 556, 544,  
1644, 544, 560, 540, 560, 540, 1652, 544, 556, 544, 556, 540, 1648, 544, 1644,
```



```
544, 556, 544, 1652, 544, 556, 544, 556, 544, 556, 540, 1648, 544, 1644, 544,
1648, 544, 1644, 544, 1648, 572, 1624, 568, 1620, 548, 1644, 544, 556, 544, 556,
544, 556, 544, 556, 540, 560, 544, 556, 544, 1648, 544, 1648, 544, 556, 544, 556,
540, 1648, 544, 556, 544, 556, 544, 1652, 544, 556, 544, 556, 544, 1644, 544,
1648, 544, 556, 544, 1648, 544, 1648, 544};
```

```
    irsend.sendRaw(irSignalz, sizeof(irSignalz) / sizeof(irSignalz[0]), khz);
}
```

```
else if ( saida < 23) { //Ajustar em 22°C
```

```
    unsigned int irSignalz[] = {4440, 4392, 564, 1628, 564, 560, 540, 1652, 540,
1652, 540, 556, 544, 556, 544, 1648, 540, 560, 540, 560, 544, 1628, 564, 556,
544, 556, 540, 1648, 544, 1648, 540, 560, 540, 1656, 540, 560, 540, 560, 540,
556, 544, 1624, 564, 1648, 544, 1624, 564, 1628, 564, 1624, 568, 1652, 544, 1648,
540, 1624, 568, 556, 544, 556, 540, 556, 544, 556, 544, 560, 540, 560, 540, 1632,
564, 1648, 540, 1648, 544, 556, 540, 1656, 540, 556, 544, 560, 540, 1628, 568,
556, 544, 556, 544, 556, 540, 1652, 544, 532, 568, 1652, 540, 1648, 544, 5236,
4444, 4380, 568, 1648, 540, 560, 540, 1652, 544, 1648, 544, 556, 540, 560, 540,
1648, 540, 564, 540, 560, 540, 1652, 544, 556, 544, 556, 540, 1652, 540, 1624,
564, 560, 540, 1632, 564, 560, 544, 556, 540, 560, 540, 1648, 544, 1648, 540,
1648, 544, 1648, 540, 1628, 564, 1632, 564, 1624, 568, 1624, 568, 556, 540, 560,
540, 556, 544, 556, 540, 560, 544, 560, 540, 1652, 544, 1648, 540, 1624, 568,
556, 540, 1656, 540, 556, 544, 556, 544, 1656, 540, 556, 544, 556, 544, 556, 540,
1656, 540, 556, 544, 1628, 564, 1652, 540};
```

```
    irsend.sendRaw(irSignalz, sizeof(irSignalz) / sizeof(irSignalz[0]), khz);
}
```

```
else if ( saida < 24) { //Ajustar em 23°C
```

```
    unsigned int irSignalz[] = {4440, 4392, 568, 1648, 544, 532, 568, 1652, 540,
1648, 544, 528, 568, 532, 568, 1648, 544, 536, 564, 536, 564, 1652, 544, 532,
564, 536, 564, 1652, 540, 1648, 540, 532, 568, 1656, 540, 536, 568, 532, 564,
532, 568, 1652, 540, 1648, 540, 1648, 544, 1652, 536, 1652, 544, 1652, 544, 1648,
540, 1648, 544, 532, 564, 536, 564, 532, 568, 532, 568, 532, 568, 536, 564, 1656,
540, 532, 564, 1632, 564, 532, 568, 1628, 564, 532, 568, 536, 564, 1632, 564,
532, 568, 1652, 544, 532, 564, 1656, 540, 532, 568, 1652, 544, 1648, 540, 5232,
4440, 4388, 564, 1624, 568, 528, 572, 1652, 540, 1652, 540, 536, 564, 532, 568,
1648, 540, 532, 572, 532, 568, 1652, 544, 532, 564, 532, 568, 1648, 544, 1648,
540, 532, 568, 1656, 540, 560, 540, 536, 564, 532, 568, 1648, 544, 1648, 540,
1648, 544, 1648, 540, 1652, 540, 1656, 540, 1648, 544, 1648, 540, 532, 568, 536,
564, 532, 564, 536, 564, 536, 564, 540, 560, 1656, 540, 532, 568, 1652, 540, 556,
544, 1652, 540, 560, 540, 536, 564, 1656, 540, 532, 568, 1628, 564, 532, 568,
1652, 540, 536, 564, 1652, 544, 1648, 540};
```

```
    irsend.sendRaw(irSignalz, sizeof(irSignalz) / sizeof(irSignalz[0]), khz);
}
```

```
else if ( saida < 25) { //Ajustar em 24°C
```

```
    unsigned int irSignalz[] = {4444, 4368, 568, 1644, 548, 552, 544, 1648, 548,
1644, 568, 532, 544, 552, 548, 1644, 544, 560, 540, 560, 544, 1648, 544, 556,
544, 556, 568, 1620, 544, 1644, 548, 552, 572, 1624, 548, 556, 544, 552, 548,
552, 572, 1620, 544, 1644, 548, 1640, 548, 1644, 572, 1620, 548, 1648, 548, 1644,
```

```
548, 1640, 548, 552, 548, 552, 548, 552, 556, 544, 544, 556, 544, 556, 544, 1648,
548, 552, 548, 552, 544, 556, 544, 1648, 548, 552, 544, 556, 548, 1648, 548, 552,
544, 1648, 548, 1644, 544, 1644, 544, 556, 544, 1652, 568, 1620, 544, 5240, 4440,
4384, 564, 1648, 544, 556, 544, 1648, 544, 1648, 544, 552, 544, 556, 544, 1644,
548, 556, 568, 532, 544, 1648, 544, 556, 544, 556, 544, 1644, 544, 1648, 544,
556, 540, 1652, 544, 560, 544, 556, 540, 556, 544, 1648, 544, 1644, 544, 1648,
540, 1648, 544, 1648, 544, 1652, 544, 1648, 540, 1648, 544, 556, 544, 532, 564,
560, 540, 532, 568, 536, 564, 532, 568, 1632, 564, 556, 540, 560, 540, 560, 540,
1652, 540, 560, 540, 560, 540, 1632, 564, 536, 564, 1652, 544, 1644, 544, 1628,
560, 536, 568, 1648, 540, 1652, 540};
```

```
    irsend.sendRaw(irSignalz, sizeof(irSignalz) / sizeof(irSignalz[0]), khz);
}
```

```
else if ( saida < 26) { //Ajustar em 25°C
```

```
    unsigned int irSignalz[] = {4444, 4380, 568, 1648, 540, 556, 544, 1652, 544,
1648, 544, 556, 540, 560, 540, 1648, 544, 560, 540, 560, 544, 1648, 544, 560,
540, 556, 544, 1648, 544, 1624, 564, 560, 540, 1652, 544, 560, 544, 556, 540,
560, 540, 1648, 544, 1624, 564, 1648, 544, 1648, 540, 1652, 568, 1628, 544, 1644,
544, 1648, 544, 556, 540, 560, 540, 556, 544, 560, 540, 560, 540, 1648, 544,
1648, 544, 556, 544, 556, 540, 556, 544, 1652, 544, 556, 540, 560, 540, 560, 544,
556, 540, 1652, 544, 1644, 544, 1648, 544, 556, 544, 1628, 564, 1648, 544, 5248,
4440, 4384, 568, 1624, 568, 552, 544, 1628, 568, 1648, 568, 532, 544, 552, 544,
1648, 544, 556, 544, 560, 540, 1652, 568, 532, 544, 556, 544, 1648, 540, 1648,
544, 556, 544, 1652, 544, 556, 548, 556, 540, 556, 544, 1648, 544, 1644, 548,
1644, 544, 1648, 544, 1648, 544, 1652, 544, 1648, 544, 1644, 548, 552, 544, 556,
544, 556, 544, 556, 544, 560, 540, 1652, 540, 1648, 544, 556, 544, 556, 544, 556,
544, 1652, 544, 556, 540, 560, 544, 556, 544, 556, 544, 1628, 564, 1648, 544,
1648, 544, 556, 540, 1652, 544, 1652, 540};
```

```
    irsend.sendRaw(irSignalz, sizeof(irSignalz) / sizeof(irSignalz[0]), khz);
}
```

```
else { //Desligar para valores acima de 26°C
```

```
    unsigned int irSignalz[] = {4408, 4432, 568, 1648, 540, 556, 544, 1652, 540,
1648, 544, 556, 544, 552, 544, 1648, 564, 536, 544, 560, 540, 1628, 564, 560,
540, 556, 568, 1600, 564, 1628, 564, 556, 540, 1632, 564, 560, 540, 1628, 584,
1628, 540, 1648, 544, 1620, 568, 556, 544, 1628, 564, 1628, 564, 1628, 568, 556,
544, 556, 540, 560, 564, 536, 540, 1624, 564, 560, 540, 560, 540, 1632, 564,
1628, 560, 1628, 564, 556, 568, 532, 544, 556, 568, 528, 544, 560, 540, 560, 540,
556, 544, 556, 544, 1624, 568, 1648, 540, 1652, 540, 1648, 540, 1624, 568, 5248,
4436, 4380, 568, 1628, 564, 556, 540, 1628, 568, 1624, 564, 560, 540, 556, 544,
1624, 564, 560, 540, 560, 540, 1628, 568, 556, 540, 560, 540, 1648, 540, 1648,
544, 556, 540, 1632, 564, 560, 540, 1652, 540, 1624, 564, 1628, 564, 1624, 564,
556, 544, 1652, 540, 1652, 540, 1656, 540, 556, 544, 556, 540, 560, 540, 556,
544, 1648, 540, 560, 540, 560, 540, 1656, 540, 1624, 564, 1628, 564, 556, 544,
556, 540, 560, 540, 556, 544, 556, 544, 560, 540, 560, 540, 556, 544, 1648, 544,
1648, 540, 1652, 540, 1648, 540, 1652, 540};
```

```
    irsend.sendRaw(irSignalz, sizeof(irSignalz) / sizeof(irSignalz[0]), khz);
}
```

```
delay(5000);
```

Anexo C

```
% Comunicaç~o Serial: L~gica Fuzzy com MATLAB e Arduino
% Mbnaliza Correa Sobreira da Silva 12/2018 - TCC UFCG */

function TCC
clear all
clc
delete (instrfind({'Port'}, {'COM5'}));
%cria a porta serial com 9600 bits por segundo
arduino_serial = serial('COM5', 'BaudRate', 9600);
%abre a porta serial
fopen (arduino_serial);
%l~ o arquivo .fis
fis = readfis ('TCC');

while(1)

%verifica in~cio do recebimento de dados
if fscanf(arduino_serial, '%d')==88
    t=fscanf(arduino_serial, '%d');
    t2 = fscanf(arduino_serial, '%d');
    IC = fscanf (arduino_serial, '%d');
    soma = fscanf(arduino_serial, '%d');
    %condiç~o para enviar o valor da sa~da
    if soma == t2+IC
        saida = evalfis([IC t2], fis);
        %envia valor da saida
        saidastr = num2str(saida, 2);
        fprintf(arduino_serial, saidastr);
        %Exibe na command window os valores de temperatura, ~ndice de
        %calor e a sa~da calculada
        disp('Temperatura Interna');
        disp(t2);
        disp('Temperatura Externa');
        disp(t);
        disp('Indice de Calor');
        disp(IC);
        disp('Saida pelo m~todo centroide');
        disp(saidastr);
    end
end

fclose (arduino_serial);
end
```