

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE
ENGENHARIA DE COMPUTAÇÃO

LEONARDO TOMASI FRANCIS

ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE PARA
INTEGRAÇÃO EFICIENTE DA ILUMINAÇÃO NATURAL E ARTIFICIAL

Araranguá
2019

LEONARDO TOMASI FRANCIS

ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE PARA
INTEGRAÇÃO EFICIENTE DA ILUMINAÇÃO NATURAL E ARTIFICIAL

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação, no curso Engenharia de Computação da Universidade Federal de Santa Catarina, Centro de Ciências, Tecnologias e Saúde do Campus de Araranguá.

Orientador: Prof. Dr. César Cataldo Scharlau

Araranguá
2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Francis, Leonardo Tomasi

ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE
PARA INTEGRAÇÃO EFICIENTE DA ILUMINAÇÃO NATURAL E
ARTIFICIAL / Leonardo Tomasi Francis ; orientador, César
Cataldo Scharlau , 2019.

84 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá,
Graduação em Engenharia de Computação, Araranguá, 2019.

Inclui referências.

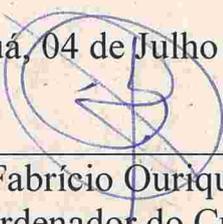
1. Engenharia de Computação. 2. Sistema de Controle de
Iluminação, Economia de Energia Elétrica, Iluminação
Artificial e Natural. I. , César Cataldo Scharlau. II.
Universidade Federal de Santa Catarina. Graduação em
Engenharia de Computação. III. Título.

Leonardo Tomasi Francis

**ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA PARA
INTEGRAÇÃO EFICIENTE DA ILUMINAÇÃO NATURAL E
ARTIFICIAL**

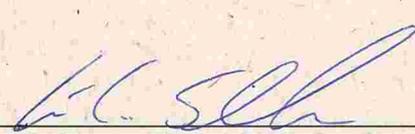
Esta trabalho de conclusão foi julgada adequada para obtenção do Título de Bacharel em Engenharia de Computação e aprovada em sua forma final pela coordenação do curso.

Araranguá, 04 de Julho de 2019.



Prof. Fabrício Ourique, Dr.
Coordenador do Curso

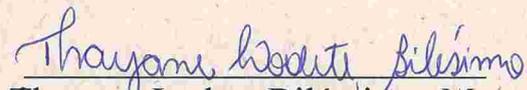
Banca Examinadora:



Prof. César Cataldo Scharlau, Dr.
Orientador



Prof. Tiago Oliveira Weber, Dr.



Thayane Lodete Biléssimo, Ma.

RESUMO

Nos últimos anos vem-se buscando encontrar alternativas que permitem racionalizar e mudar hábitos a respeito do uso de energia elétrica. Estudos da *Internacional Energy Agency* (IEA) indicam o aumento da demanda de energia elétrica nos próximos anos, o que reforça a necessidade de medidas para reduzir o desperdício. Tendo em vista que nas edificações os gastos com iluminação artificial representam expressivos valores no consumo de energia, busca-se encontrar alternativas que contribuam para a economia. O objetivo desse trabalho é o desenvolvimento de sistemas de controle e validação experimental para um ambiente em escala de uma sala de aula da Universidade Federal de Santa Catarina - Campus Araranguá, de modo a realizar a integração entre iluminação natural e artificial. Para atender os objetivos, serão inseridos no ambiente em escala um conjunto de sensores, que serão responsáveis pela aquisição de valores de iluminância do ambiente, fornecendo informações para um sistema de controle, que deve ajustar o nível de iluminação artificial de forma adequada. Com este trabalho, buscou-se analisar o desempenho do sistema aplicado e a viabilidade técnica dessa alternativa. Assim, o cenário de uma área representou redução de uso de potência em 75,76% da potência total, enquanto com quatro áreas representou redução de 78,65%. O cenário utilizando duas áreas controladas por controladores PI representou melhor resultando, 88,22% da redução em potência utilizada. Na aplicação do melhor cenário encontrado em uma sala de aula real, existe a possibilidade de integrar outras alternativas, como por exemplo o sensor de presença.

Palavras-chave: Sistema de Controle de Iluminação, Economia de Energia Elétrica, Iluminação Artificial e Natural.

ABSTRACT

In recent years, solutions in order to ration and change habits regarding the use of electricity power are increasingly being searched. The need to reduce energy consumption has been reinforced due to the increased demand in the next few years studied by IEA (International Energy Agency). Solutions which contribute in energy savings are being searched, given that buildings account a significant value of total energy consumption. This work aimed at developing a control system and an experimental validation of a scaled environment based on a classroom from the Federal University of Santa Catarina - Araranguá Campus, in order to ensure that the natural and artificial illumination will be integrated efficiently. To accomplish the objectives and results a set of lighting sensors will be inserted into the scaled environment, which also will be responsible for the acquisition of luminance values measured in the environment, thus feeding the control system with data, as a result it will dimmer the artificial lighting. By the end of this work, it is aimed to analyze the performance of the applied system and the technical feasibility of the developed alternative. The scenario with one area have reached power usage reduction by 75,76% of total power, while four areas scenario have reached 78,65%. The best result performed was with two areas scenario controlled by PI controller, it has reached power usage reduction of 88,22%. Applying the same scenario in a real room it has the possabilitie to integrate with other solutions, for exemple presence sensors, increasing the energy saving.

Keywords: Lighting Control System, Electrical Energy Saving, Artificial and Natural lighting.

AGRADECIMENTOS

Agradeço a Deus e Meishu-Sama, por terem me proporcionado oportunidades e proteção durante a graduação.

Agradeço a meus pais por sempre terem me proporcionado acesso a educação. Especialmente minha mãe, Liette Fátima Tomasi, pelo apoio e motivação durante toda graduação, acreditando no meu potencial.

Agradeço ao meu orientador, César Cataldo Scharlau, pela dedicação a profissão que me serviu de inspiração acadêmica e pelas instruções que foram imprescindíveis para execução desse trabalho.

Agradeço aos professores, mestres e doutores, que se dispuseram a passar seu conhecimento e me instigaram a continuar buscando conhecimento.

Agradeço aos meus amigos e colegas que fizeram parte de todo esse processo de grande aprendizado, pelo apoio e ajuda a entender quando eu não entendia, e que fizeram parte dessa experiência incrível na minha vida.

E agradeço, por fim, ao William dos Passos pela companhia, apoio e motivação.

"Setting goals is the first step in turning the invisible into the visible."

Robbins, Tony

LISTA DE ILUSTRAÇÕES

Figura 1 – Luxímetro Minipa-1020	17
Figura 2 – Fenômenos Físicos da Iluminação	18
Figura 3 – Estratégias de aproveitamento na iluminação	20
Figura 4 – Sensor LDR	22
Figura 5 – Planta Baixa do Ambiente Real	23
Figura 6 – Ambiente em escala	24
Figura 7 – Placa de Circuito Impresso	26
Figura 8 – Esquemático do circuito de aquisição por LDR	27
Figura 9 – Esquemático do acionamento fitas das fitas de LED	28
Figura 10 – PCI, Arduino e Módulo RTC	29
Figura 11 – Diagrama de blocos do processo de medição	31
Figura 12 – Área representante de 1 área	32
Figura 13 – Área representante de 2 áreas	33
Figura 14 – Área representante de 4 áreas	33
Figura 15 – Espectro de frequência - cenário uma área	36
Figura 16 – Entrada de dados com e sem filtro - 1 Área	37
Figura 17 – Espectro de frequência - Cenário de duas áreas	38
Figura 18 – Entrada de dados sem filtro e filtro passa-baixa 2 Áreas	39
Figura 19 – Espectro Frequência de 4 Áreas	40
Figura 20 – Entrada de dados sem filtro, filtro passa-baixa e filtro exponencial - 4 Áreas	41
Figura 21 – Valores Medidos e Aproximação - Sensor 0	43
Figura 22 – Valores Medidos e Aproximação - Sensor 1	43
Figura 23 – Valores Medidos e Aproximação - Sensor 2	44
Figura 24 – Valores Medidos e Aproximação - Sensor 3	44
Figura 25 – Cenários 1 Área - Comparação de utilização em porcentagem	47
Figura 26 – Cenários 4 Áreas - Comparação de utilização em porcentagem	48
Figura 27 – Cenários 2 Áreas - Comparação de utilização em porcentagem	49
Figura 28 – Cenários 2 Áreas - Bloqueio da porta e quadro-negro	49

LISTA DE TABELAS

Tabela 1 – Relação PCI e Arduino	28
Tabela 2 – Dados dos Sensores A0, A1, A2, A3	42
Tabela 3 – Tensão, Corrente e Potência das áreas A0, A1, A2, A3 - Potência Máxima	46
Tabela 4 – Tensão, Corrente e Potência das áreas A0, A1, A2, A3 - <i>Dimmer</i> . .	46

LISTA DE SIGLAS

A	Ampere
ABNT	Associação Brasileira de Normas Técnicas
BACS	<i>Building Automation & Control System</i>
BMS	<i>Building Management Systems</i>
IDE	Ambiente de Desenvolvimento Integrado, do inglês <i>Integrated Development Environment</i>
IEA	<i>International Energy Agency</i>
Inmetro	Instituto Nacional de Metrologia, Qualidade e Tecnologia
IoT	Internet das Coisas, do inglês <i>Internet of Things</i>
k	Kilo
LDR	<i>Light-Dependent Resistor</i>
LED	<i>Light Emitting Diode</i>
lm	Lumen
lx	Lux
m	Metro
NPN	Negativo-Positivo-Negativo, do inglês <i>Negative-Positive-Negative</i>
PCI	Placa de Circuito Impresso
PWM	<i>Pulse Width Modulation</i>
R	Resistência
RTC	Relógio de Tempo Real, do inglês <i>Real Time Clock</i>
TWh	Terawatt-hora
V	Volt
W	Watt

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.1.1	Geral	14
1.1.2	Específicos	14
1.2	Justificativa	15
1.3	Organização do Trabalho	15
2	ILUMINAÇÃO NAS EDIFICAÇÕES	16
2.1	Conceitos Básicos	16
2.2	Iluminação Natural e Artificial	18
2.3	Eficiência Luminosa	19
2.4	Sistemas Eficientes	20
2.4.1	Medição por Sensores	22
3	SISTEMA DE CONTROLE DA ILUMINAÇÃO ARTIFICIAL	23
3.1	Descrição	23
3.2	Projeto de <i>Hardware</i>	25
3.2.1	Placa de Circuito Impresso	25
3.2.1.1	Modelo esquemático	26
3.2.2	Arduino Mega	27
3.2.3	Materiais utilizados	28
3.3	Projeto de <i>Software</i>	29
3.3.1	<i>Software</i> Base	30
3.3.2	Projeto dos Filtros	31
3.3.3	Mapeamento para conversão lx	32
3.3.4	Cenários Implementados	32
3.3.4.1	Uma Área	32
3.3.4.2	Duas Áreas	33
3.3.4.3	Quatro Áreas	33
3.3.5	<i>Softwares</i> Utilizados	34
4	RESULTADOS	35
4.1	Testes e Configuração	35
4.1.1	Filtros	35
4.1.1.1	Uma Área	35
4.1.1.2	Duas Áreas	37
4.1.1.3	Quatro Áreas	39

4.1.2	Função Matemática de Conversão para Ix	42
4.1.3	Controlador PI	45
4.1.4	Potência das fitas de LED	45
4.2	Resultados em Malha Fechada	46
5	CONSIDERAÇÕES FINAIS	50
5.1	Trabalhos Futuros	51
	REFERÊNCIAS	52
	Apêndice A	55
	Apêndice B	58
	Apêndice C	61
	Apêndice D	64
	Apêndice E	71
	Apêndice F	78

1 INTRODUÇÃO

Nos últimos 25 anos a demanda de energia elétrica vem crescendo e superando as perspectivas neste setor. Visto que existem alguns tipos de combustíveis que geram energia, como por exemplo o gás, o óleo, carvão e entre outras como a energia elétrica, que por sua vez representa 19% do consumo mundial (IEA, 2018c). Países desenvolvidos tem uma taxa de crescimento de consumo de energia de 1,1%, enquanto que em países emergentes esse índice pode chegar a 3,2% (PÉREZ-LOMBARD; ORTIZ; POUT, 2008).

Segundo os dados do anuário mundial de energia (em inglês *World Energy Outlook*) publicados pela IEA (2018a), prevê-se que até 2040 a geração de energia elétrica passará de 24 para 40 TWh. Estima-se que entre 30% a 40% do total de energia elétrica consumida global é por edifícios, domésticos e não domésticos (Comissão Europeia, 2003).

Uma das causas do aumento da demanda de eletricidade nas últimas décadas é o crescimento populacional nos centros urbanos. Em 1973, devido à crise do petróleo, notou-se que era melhor economicamente efficientizar o consumo do que aumentar a oferta de energia elétrica. O conceito de eficiência energética nas edificações busca criar ambientes com condições térmicas, visuais e acústicas adequadas para os usuários com baixo consumo de energia (LAMBERTS; DUTRA; PEREIRA, 2014)[p.5].

A eficiência energética sempre foi um assunto de interesse. No entanto, nos últimos anos muitos países da UE - União Europeia, apoiados por políticas públicas, vem buscando novas técnicas para tornar o consumo de energia elétrica mais eficiente (Comissão Europeia, 2003).

No Brasil, a eficiência energética dos edifícios é mensurada a partir do processo de etiquetagem. Nesse processo são considerados o consumo de energia, o conforto térmico dos usuários do ambiente e características físicas como envoltória, condicionamento de ar, iluminação natural e iluminação artificial (MINISTÉRIO DO MEIO AMBIENTE (MMA), 2015). Considera-se iluminação natural aquela proveniente, diretamente ou indiretamente, de fontes de luz como o sol, céu ou refletida por superfícies (LAMBERTS; DUTRA; PEREIRA, 2014). Por sua vez, a iluminação artificial aquela provenientes de fontes artificiais como lâmpadas, laminários e pendentés.

A forma, orientação e até as cores aplicadas nas edificações interferem no uso eficiente da luz natural, interferindo no consumo de iluminação artificial (DIDONÉ; OSCAR; PEREIRA, 2009). Segundo dados publicados na revista Lume por Souza

(2008), o uso correto da luz natural permite alcançar uma economia de até 50% com energia elétrica gasta com iluminação (SOUZA, 2008).

Atualmente, 7% do consumo global em energia elétrica é gasto com iluminação artificial. Esse valor reduzido se deve as políticas governamentais que incentivam a diminuição do consumo. Além disso, a evolução tecnológica dos equipamentos também tem grande influência na redução do consumo (IEA, 2018b). Os gastos no Brasil são de 14% para o setor residencial, 22% para o setor comercial e 23% para os edifícios públicos (LAMBERTS; DUTRA; PEREIRA, 2014).

A previsão é que em 2040 os edifícios se tornarão até 40% mais eficientes do que atualmente. Essa melhoria se dará devido a inclusão de diversas estratégias eficientes como a utilização de sistemas de controle, agendamento, sensores de ocupação, zoneamento de áreas e demais técnicas de automação para prédios como BMS (*Building Management Systems*) e BACS (*Building Automation & Control System*) que potencializam a economia de energia (IEA, 2018a) (GUL; PATIDAR, 2015).

Dentro desse contexto, o presente trabalho se propõe a analisar, em um ambiente em escala, a aplicação de técnicas de controle para integração eficiente do uso da luz natural e artificial. Serão comparados diferentes cenários, analisando o impacto no consumo de energia elétrica, através de resultados experimentais.

1.1 Objetivos

Nesta seção serão apresentados os objetivos geral e específicos contemplados neste trabalho.

1.1.1 Geral

Desenvolver um ambiente em escala, a partir de dados de um ambiente real, e aplicar um sistema de controle de iluminação buscando integrar de maneira eficiente o uso da luz natural e artificial para diminuir o consumo de energia elétrica.

1.1.2 Específicos

Para alcançar o objetivo geral serão necessários atingir os seguintes objetivos específicos:

1. Desenvolver um protótipo com as características da sala em escala para viabilizar a execução dos testes;
2. Implementar um sistema capaz de monitorar os níveis de iluminância dentro do ambiente em escala;
3. Aplicar técnicas para controlar os níveis de iluminância através do acionamento de um conjunto de LEDs - *Light Emitting Diode* (Diodo Emissor de Luz);
4. Analisar e comparar o desempenho das alternativas de controle propostas.

1.2 Justificativa

Considerando que, mundialmente, 7% da energia elétrica produzida é utilizada para iluminar ambientes (IEA, 2018b), no Brasil, esse valor alcança 22% para o setor comercial e 14% para o setor residencial (LAMBERTS; DUTRA; PEREIRA, 2014). A Universidade Federal de Santa Catarina no Campus Araranguá do bairro Jardim das Avenidas gasta 32% em iluminação artificial (BILÉSIMO, 2015), por isso, vem-se buscando encontrar soluções para a racionalização desse uso.

Segundo dados do PROCEL (2002), o uso da luz natural nos ambientes pode trazer uma redução de até 50% do consumo de energia com iluminação. Soluções tecnológicas, como sistemas de automação, estão diretamente ligados à economia de energia. Entre as existentes, o controle de iluminação pode proporcionar até 40% (NIERO, 2017).

1.3 Organização do Trabalho

A estrutura do trabalho está organizada em 5 capítulos com os assuntos respectivos, incluindo neles, o presente capítulo de Introdução.

O **capítulo 2** refere-se a Iluminação nas Edificações e trata de definições e conceitos importantes de iluminação para as edificações.

O **capítulo 3** refere-se ao Sistema de Controle da Iluminação Artificial e trata-se da metodologia utilizada para o desenvolvimento do protótipo em escala, *hardware* e *software* presentes neste trabalho.

O **capítulo 4** se refere aos Resultados e trata dos testes e os resultados encontrados a partir dos testes.

O **capítulo 5** se refere as Considerações Finais e análises realizadas com a execução deste trabalho.

O apêndice apresenta os códigos-fonte em malha aberta e fechada utilizados para a execução dos testes presentes neste trabalho.

2 ILUMINAÇÃO NAS EDIFICAÇÕES

Neste capítulo serão apresentadas as definições e os conceitos básicos sobre iluminação utilizados neste trabalho.

2.1 Conceitos Básicos

Dentro de projetos de ambientes é essencial a aplicação de conceitos de conforto visual devido a sua influência no desempenho das tarefas executadas pelos usuários no ambiente (LAMBERTS; DUTRA; PEREIRA, 2014).

Para Lamberts, Dutra e Pereira (2014) o conforto visual é um conjunto de condições que permitem as pessoas realizarem tarefas visuais com precisão e assertividade, sem esforços visuais e minimizando possibilidade de riscos de acidentes causados por interferências na visão.

É definido como fluxo luminoso (ϕ_v) a quantidade de luz emitida por uma fonte luminosa por segundo em todas as direções, possui unidade Lumen (lm). A relação de fluxo luminoso que incide sob o metro quadrado de uma área é definido por iluminância (E), possui unidade Lux (lx) (PROCEL, 2011).

$$E = \frac{\phi_v}{S} \quad (2.1)$$

O processo de medição de nível de iluminância pode ser realizado por um aparelho denominado luxímetro. A Figura 1 apresenta um exemplo desse tipo de equipamento. Para realizar a verificação experimental das condições de iluminância, tanto de ambientes reais como ambientes em escala, são estabelecidos procedimentos e parâmetros na norma NBR 15215-4 ABNT (2004). Como exemplos desses procedimentos e parâmetros, é possível citar:

1. Construir modelos em escala não maior do que 1:40;
2. Adequar as refletividades das superfícies e representar o mais corretamente possível as refletividades das superfícies reais;
3. Evitar modelos mal executados onde possam ocorrer vazamentos de luz em suas juntas;
4. Modelar adequadamente os detalhes das aberturas.

A antiga norma NBR 5413 ABNT (1992) determinava os níveis de iluminação requeridos para cada atividade por valor baixo, médio e alto de iluminância. A partir de 2013, a inclusão da NBR ISO/CIE 8995-1 substituiu a antiga norma. Na normal atual

tem-se apenas uma iluminância média para cada atividade. Para uma sala de aula por exemplo, é utilizado 500 lx (ABNT, 2013).

Figura 1 – Luxímetro Minipa-1020



Fonte: Página da Minipa em Luxímetros.

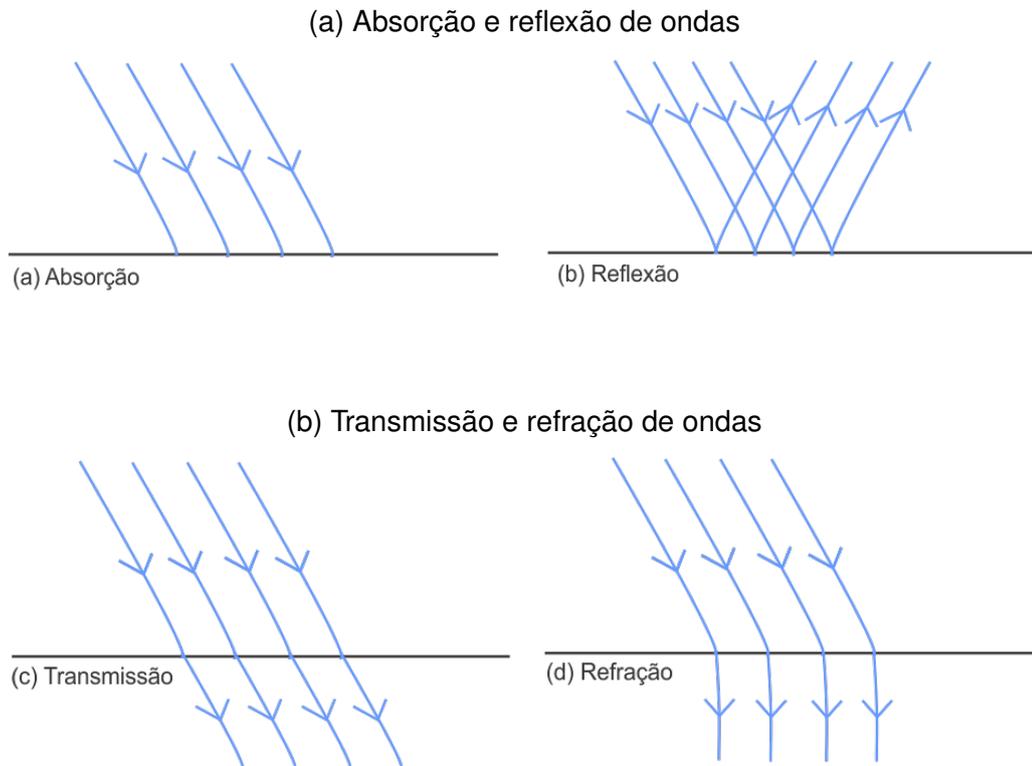
Existem alguns fenômenos físicos aplicáveis para os fluxos luminosos, como a absorção, a reflexão, a refração e a transmissão, conforme demonstrado na Figura 2. Segundo os dados de PROCEL (2011) a absorção acontece quando raios de luz que incidem em uma superfície que são absorvidos por ela. A reflexão acontece quando esses raios são refletidos de volta para o meio de origem.

O fenômeno da reflexão pode acontecer de forma especular quando está refletido em apenas uma direção ou difusa quando está refletindo para diversas direções. Já a transmissão acontece quando os raios penetram no meio. A refração acontece ao ser transmitida de um meio para outro, alterando a velocidade e o comprimento de onda. Por esse motivo, ocorre um desvio de direção em relação à original (PROCEL, 2011).

Quando se fala em iluminação é comum estudar alguns problemas relacionados. O ofuscamento, por exemplo, ocorre devido a variação rápida e alta de iluminâncias (LAMBERTS; DUTRA; PEREIRA, 2014). Ao se olhar por uma janela durante um dia muito ensolarado pode ocorrer o ofuscamento direto. Esse efeito acontece quando a fonte de luz está no campo de visão. O ofuscamento indireto, por sua vez, acontece quando a fonte de luz está em uma direção diferente do campo de visão. Dependendo da incidência de luz, o ofuscamento pode causar desconforto visual aliado com o enfraquecimento da visão dos objetos, que resultam em fadiga visual (LAMBERTS; DUTRA; PEREIRA, 2014).

Segundo Silva (2016), para ter um ambiente com boa iluminação é importante controlar o ofuscamento, brilho, reflexos e evitar excessos, já que nível alto de

Figura 2 – Fenômenos Físicos da Iluminação



Fonte: Autor.

iluminância não significa, necessariamente, conforto visual.

2.2 Iluminação Natural e Artificial

A iluminação está diretamente ligada com o conforto dos usuários, permitindo harmonizar e valorizar o ambiente. Além disso, possui grande influência na produtividade e rendimento das pessoas (SILVA, 2016). Estudos conduzidos por Begemann, S.H.A. et al (1994 apud De Carli; De Giuli; ZECCHIN, 2008) apontam que ambientes com má iluminação podem causar problemas de sono, perda de produtividade e até depressão.

Para Lamberts, Dutra e Pereira (2014) as fontes de luz natural são o sol, o céu e as superfícies, edificadas ou não, que fornecem respectivamente luz direta, luz difusa, luz refletida ou indireta. A iluminação de ambientes está associada com a utilização de fontes luminosas naturais e artificiais que buscam garantir visibilidade, segurança e orientação dentro de ambientes (WENDEN, 2013).

Devido a variação de iluminação natural durante o dia, entre o plano de trabalho e as aberturas é necessária a inclusão das fontes de luz artificial, permitindo a realização de atividades sem limitação de horário. Segundo Lamberts, Dutra e Pereira (2014) na arquitetura de ambientes se analisa muito a relação entre iluminação

natural e iluminação artificial. Sabe-se que a integração traz diversos benefícios e uma complementa a outra. Os benefícios são diversos com a aplicação correta da iluminação de um ambiente, como aumento de produtividade, melhora na concentração e harmonia ao ambiente (SILVA, 2016).

Silva (2016) enfatiza que os ambientes comerciais devem ser projetados levando em consideração os níveis de iluminância adequados para cada atividade. Além disso, Lamberts, Dutra e Pereira (2014) expõem que não se deve desprezar o uso da iluminação natural já que o contato com o exterior ajuda a aliviar o estresse.

2.3 Eficiência Luminosa

Por muitos anos, os arquitetos estudaram formas de tornar os ambientes mais agradáveis e mais eficientes em termos de iluminação. Por esse motivo, sabe-se que existem diversas alternativas que auxiliam o aproveitamento da iluminação natural nos ambientes. Algumas estratégias são a utilização de iluminação lateral, zenital horizontal, angular, solar direta, indireta e átrio (ROBBINS, 1986 apud FONSECA, 2007). Exemplos de estratégias são mostrados na Figura 3.

O uso mais comum é a utilização da iluminação lateral, vista na Figura 3(a). Porém, de acordo com as necessidade locais, outras estratégias podem ser adotadas visando melhorar o aproveitamento da luz natural.

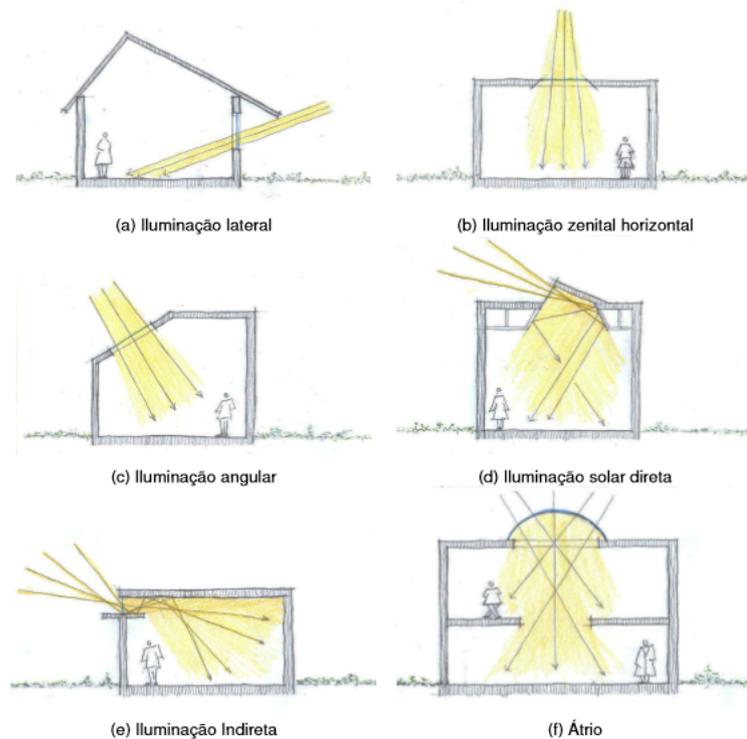
Para Souza (2008) o uso de estratégias buscando aproveitamento da iluminação natural pode trazer uma economia de até 50% nos gastos com iluminação nas edificações. Por esse motivo, a inclusão desses detalhes nos projetos de ambientes se torna cada vez mais importante a longo prazo.

As cores internas também desempenham um papel importante na eficiência dos ambientes. Por exemplo, cores claras tem maior capacidade de refletir a luminosidade, enquanto as cores escuras em absorver ela. Por esse motivo, ambientes com tons claros auxiliam na reflexão da luminosidade natural e artificial, trazendo uma melhor distribuição da luminosidade no ambiente.

O uso da iluminação artificial é, sem dúvida, necessário e importante nos ambientes. Porém, a integração entre iluminação artificial e natural deve ser balanceada, buscando a minimização do uso da iluminação artificial, mantendo conforto visual e trazendo economia de energia elétrica.

Todos os componentes que integram os sistemas de iluminação natural e artificial devem ser analisados conjuntamente, inclusive as lâmpadas presentes no sistema de iluminação. No início de 2001 houve um período de apagões no Brasil, o que motivou o Inmetro - Instituto Nacional de Metrologia, Qualidade e Tecnologia a conscientizar os consumidores sob os problemas do uso das lâmpadas incandescentes (GANDRA, 2016).

Figura 3 – Estratégias de aproveitamento na iluminação



Fonte: Fonseca (2007) [p. 26]

Segundo os dados de PROCEL (2011) a eficiência luminosa é definida como a relação entre fluxo luminoso e a potência, com a unidade de lm/W . Esse aspecto é importante a ser considerado, pois quanto maior seu valor, mais econômica será em termos de consumo de energia elétrica. As lâmpadas incandescentes geralmente possuem eficiência luminosa de 10 lm/W a 15 lm/W , as fluorescentes comuns 55 lm/W a 75 lm/W e, segundo Gul e Patidar (2015) é possível encontrar lâmpadas LED com eficiência de 113 lm/W .

A substituição das lâmpadas incandescentes por fluorescentes permite uma economia de até 75%. As lâmpadas LED possibilitam uma economia de 85% se comparadas com as lâmpadas incandescentes. Por esse motivo, houve uma restrição do uso de lâmpadas incandescentes a partir da Portaria Interministerial 1.007/2010 buscando racionalizar o consumo de energia elétrica (GANDRA, 2016).

2.4 Sistemas Eficientes

Atualmente, existem diversos sistemas utilizados em edificações que buscam a diminuição do consumo de energia elétrica. A preocupação se baseia no tamanho dos ambientes e os desperdícios que podem ocorrer. Assim, são implementados sistemas inteligentes que auxiliam no controle e, por consequência, na economia de energia elétrica.

Normalmente, as terminologias sobre o tema de automação de edifícios são BMS - *Buildings Management System* (Sistema de Gerenciamento de Edifícios), BCS - *Buildings Control System* (Sistema de Controle para Edifícios), BACS - *Building Automation and Control Systems* (Sistemas de Controle e Automação de Edifícios), também denominados BAS - *Buildings Automation System* (Sistemas de Automação para Edifícios). Além disso, os BEMS - *Building Energy Management Systems* (Sistemas de Gerenciamento de Energia em Edifícios) normalmente são utilizados para medição de consumo de energia elétrica em edificações (P. Palensky, 2003).

Em 1970 se iniciaram as primeiras implementações dos BMS. Com os constantes avanços tecnológicos, hoje é possível realizar controle, monitoramento, automação e otimização de todos os parâmetros de um edifício (TARIQ et al., 2012). Ainda segundo Tariq et al. (2012) os BMS permitem o controle do sistema de iluminação, condicionamento de ar, elevadores, câmeras de segurança e até alarmes de incêndios, atuando através de sensores e informações digitais que permitem realizar um controle inteligente e que possibilita a economia de energia elétrica.

Além destes sistemas complexos, é possível incluir alternativas mais simples que já contribuem na economia. Por exemplo, os sensores estão cada vez mais acessíveis. Os sensores de presença detectam a movimentação e presença de pessoas no ambiente, assim, mantendo a iluminação artificial do ambiente ligada enquanto for necessária. No estudo conduzido por Petry et al. (2009), a inclusão de sensores de presença no sistema de iluminação de duas salas de aula permitiu uma economia de 28%.

Existem temporizadores que permitem que a iluminação artificial fique ligada por tempo determinado. Após esse tempo, a iluminação volta a ficar apagada. Com esse processo de liga e desliga, há uma economia considerável, já que há ambientes que não precisam ficar com a iluminação ligada 24 horas por dia.

O uso de sistemas de controle para eficientizar o uso da iluminação artificial é um campo de constante estudo. Silva e Taschetto (2014) conseguiram uma economia de até 22,28% mantendo as condições de trabalho no ambiente de um laboratório de sistemas térmicos. Os resultados de Byun e Shin (2018) foram de 43% de economia. A aplicação de sistemas de controle de iluminação desenvolvida considerava parâmetros de satisfação e padrão de usuários.

Segundo os estudos de Yang, Chang e Chou (2016) aplicados em um estacionamento, a troca de lâmpadas fluorescentes com uma eficiência luminosa maior permitiu uma economia de 65%. A integração dessas lâmpadas com o sistema de controle de iluminação conseguiu atingir um total de 77,5% de economia de energia elétrica.

A dimerização (*dimming*) está associada a alteração do nível de iluminância de um sistema de iluminação automatizado ou não que permite variar de 0% a 100%.

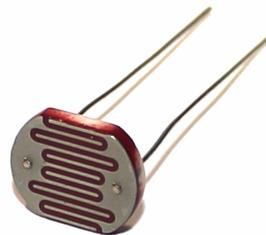
Analogamente, pode-se relacionar com o aumento e diminuição do volume de um alto falante. A utilização desta técnica auxilia também na redução do consumo de energia elétrica. Existem situações durante o dia em que não se faz necessário a utilização da potência total das lâmpadas. Com o ajuste, é possível diminuir o consumo e tornar o sistema mais eficiente.

O estudo da eficiência de sistemas de iluminação envolve a integração de diversas tecnologias e o resultado traz benefícios econômicos e ambientais. Busca-se cada vez mais minimizar os desperdícios e maximizar a eficiência de uso.

2.4.1 Medição por Sensores

Atualmente, a utilização de sensores está cada vez mais favorável ao desenvolvimento de soluções tecnológicas. Comercialmente, existem diversos componentes que realizam a medição da iluminância. Porém, quanto melhor a resolução e mais preciso, maior o seu custo.

Figura 4 – Sensor LDR



Fonte: Sensor LDR

Os sensores de luminosidade LDR (*Light-Dependent Resistor*) ou fotoresistores (Figura 4) são capazes de medir a intensidade de luz do ambiente por meio da variação da resistência interna. Normalmente são mais utilizados devido ao seu preço. Atualmente, a unidade pode chegar a R\$0,30.

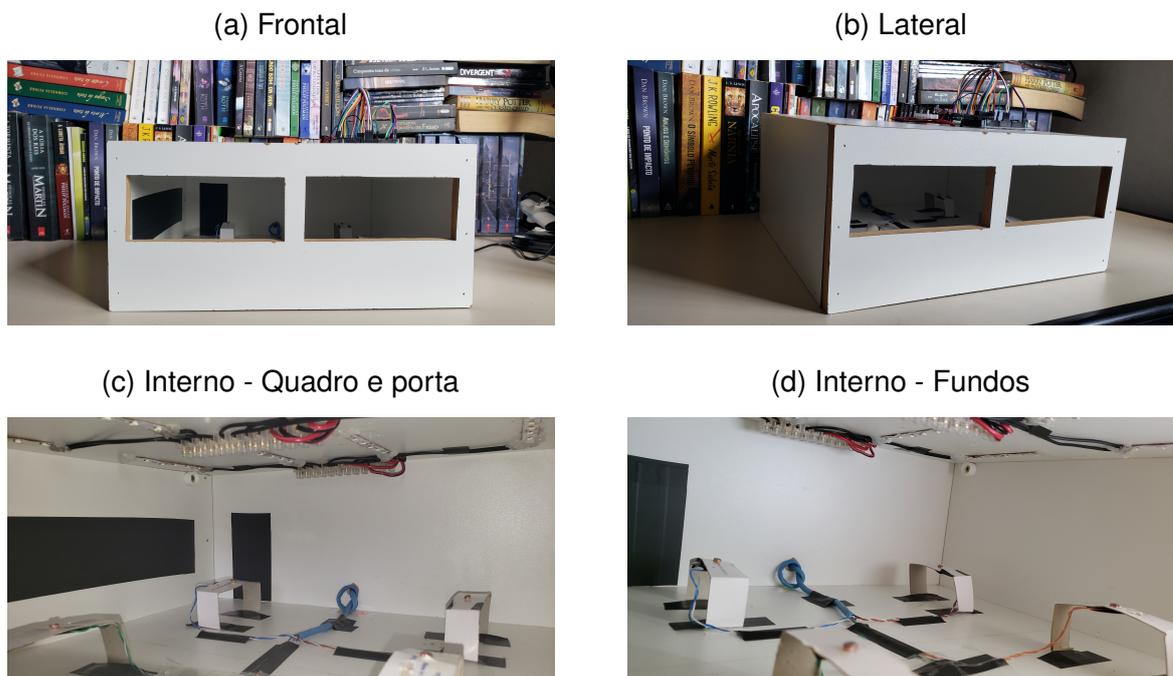
Além dos fotoresistores existem sensores digitais com resolução de AD de 16 bits. O sensor BH1750FVI possui faixa de medição de 1 a 65.535 lx e o sensor TSL2561 com faixa de 0,1 à 40.000 lx. O preço da unidade desses sensores respectivamente pode variar de R\$16,90 a R\$53,90.

iluminação natural em edifícios. Esses modelos ajudam na simulação de fenômenos físicos que podem ser avaliados quantitativamente ou qualitativamente.

Nesse trabalho, a motivação da criação do ambiente em escala considerou a viabilidade econômica para o desenvolvimento, facilidade para compra de equipamentos e simplificação na solução a ser aplicada para realização dos testes. Além disso, o uso de um modelo em escala permite realizar testes sem afetar a rotina de uso de uma sala de aula.

Inicialmente, a base do protótipo foi dividida em quatro áreas e foram instalados sensores de luminosidade do tipo LDR (*Light-Dependent Resistor*) em cada uma dessas áreas. Essas divisões servem para avaliar os níveis de iluminância nas partes mais próximas e afastadas das janelas, conforme mostra Figura 6c e 6d. Cada cor na Figura 5 representa uma área, sendo: azul a Área 0 (A0), verde a Área 1 (A1), amarelo a Área 2 (A2) e vermelho a Área 3 (A3). No interior do ambiente em escala, foram representados com fita escura o espaço onde fica o quadro-negro e a porta.

Figura 6 – Ambiente em escala



Fonte: Autor.

Os valores de iluminância lidos pelos sensores devem ser convertidos em sinais elétricos e enviados para o canal analógico da placa de desenvolvimento Arduino MEGA. O conjunto de informações desses quatro sensores posteriormente devem servir de entrada para o sistema de controle.

Os dados de entrada irão sofrer calibração através de uma função matemática definida para cada sensor. A função foi definida e calibrada com base nos valores

lidos pelo Luxímetro Minipa, modelo MLM-1020. Os valores de entrada também são filtrados, pois em alta frequência existe variação devido a existência de ruído. Nesse trabalho foram avaliados dois tipos de filtros digitais: passa-baixa de primeira ordem e exponencial.

Serão analisados os seguintes cenários:

- Uma área de sensores, o controle é feito com base na média dos quatro sensores A0123(S0, S1, S2 e S3);
- Duas áreas de sensores, o controle é realizado a partir da média de dois grupos de sensores A01(S0 e S1) e A23(S2 e S3);
- Quatro áreas de sensores, com controle independente para cada área A0(S0), A1(S1), A2(S2) e A3(S3).

Como estratégia de controle, foi utilizado o controlador PI (Proporcional-Integral) cujos parâmetros Proporcionais (K_p), Integrais (K_i) foram escolhidos de forma experimental. A equação (3.1) representa o controlador PI, onde $u(t)$ é definido como a saída em relação ao tempo, $e(t)$ como a entrada menos o erro em relação ao tempo, K_p e K_i são constantes (BEZERRA, 2007).

$$u(t) = K_p.e(t) + K_i. \int_0^t e(\tau)d\tau \quad (3.1)$$

O controlador PI irá atuar diretamente sob as fitas LED, aumentando ou diminuindo a intensidade para manter um valor de referencial. No trabalho em questão não foi utilizada nenhuma prática de evitar ofuscamento como as prateleiras de sol e brises horizontais e verticais. As cores claras internas do protótipo estão representadas de acordo com o ambiente real, e auxilia na reflexão da luz natural e artificial.

3.2 Projeto de *Hardware*

Nesta seção será detalhado o projeto do *hardware* utilizado para execução deste trabalho.

3.2.1 Placa de Circuito Impresso

O projeto da Placa de Circuito Impresso (PCI) foi desenvolvido para permitir o controle com alguns cenários possíveis: o controle automático, controle manual, projeção e ensino, que podem ser acionados por botões. Foram incluídos LEDs que permitem informar a seleção dessas opções. O modelo real e o layout realizado podem ser conferidos na Figura 7.

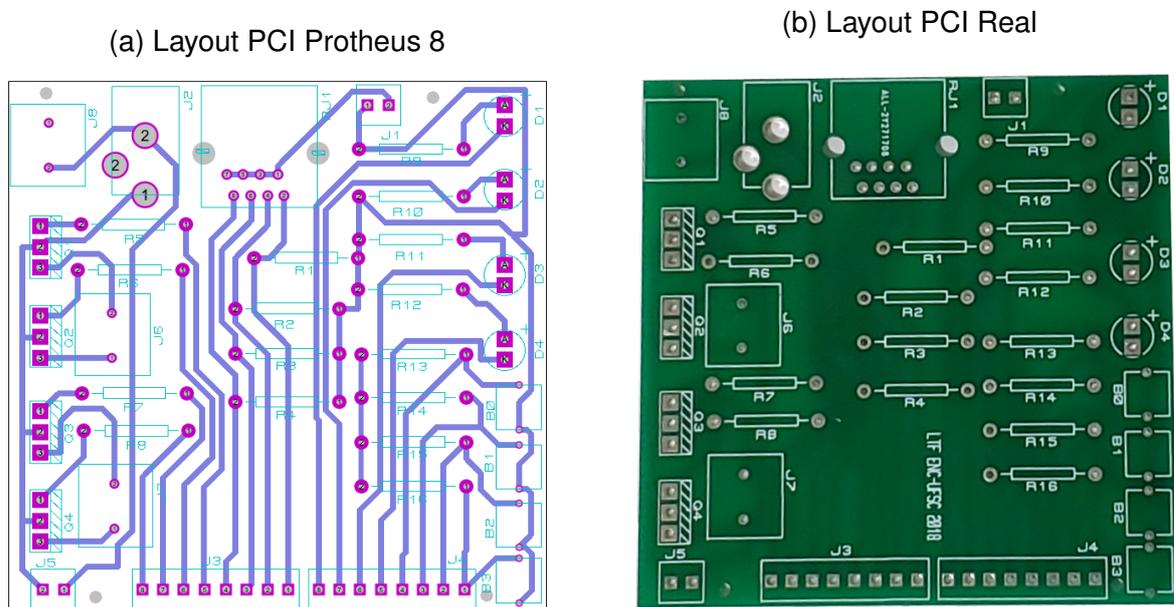
O cenário de controle automático deve realizar o processo de regulação de iluminação do ambiente, buscando convergir para o valor de iluminância de referência. O controle manual foi projetado para realizar o ajuste manual da iluminação com auxílio dos botões, sem passar pelo processo do sistema de controle. Os modos projeção

e ensino são cenários pré-programados que permitiriam diminuir a iluminação para realizar a projeção em sala de aula ou aumentar a luminosidade interna da sala acima de 500 lx.

A aquisição dos sinais foi feita pelo sensor LDR, que identifica a variação de luminosidade a partir da sua resistência interna. A interligação dos sensores à placa é feita através de um conector RJ-45. A comunicação com a placa Arduino Mega é feita através do conector J3. A escolha da utilização do conector RJ-45 foi devido a quantidade de sensores presentes, dessa forma, caso existisse uma quantidade maior de sensores seria necessário utilizar outro tipo de conector.

O sistema de processamento de dados e controle é implementado no Arduino, que aplica sinais PWM na base dos transistores que então acionam as fitas de LED presentes no teto do ambiente em escala. Através do conector J2, as fitas de LED são alimentadas por uma fonte externa com uma tensão máxima de 12 V. De acordo com o valor de tensão que é aplicado, as fitas são acionadas com sua iluminância ajustada pelo sistema de controle. Os sensores detectam a variação de iluminância, e após receber novos valores, o processo continua buscando manter um valor de referência.

Figura 7 – Placa de Circuito Impresso



Fonte: Autor.

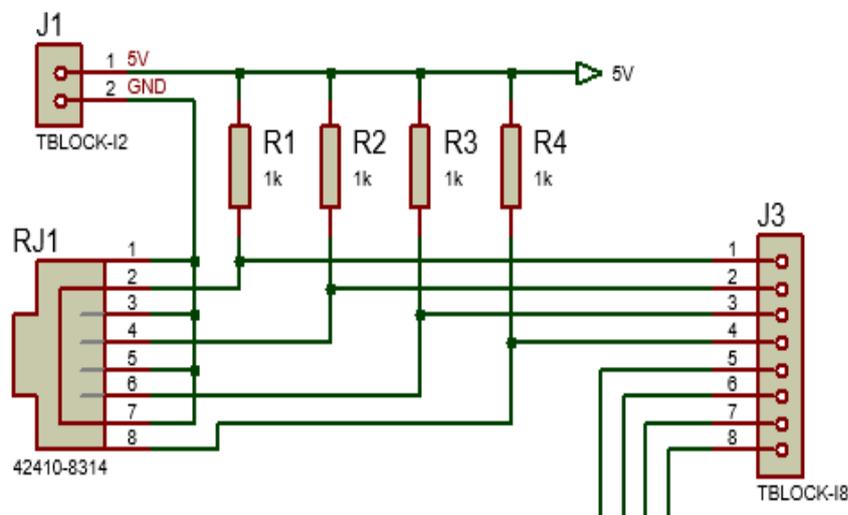
3.2.1.1 Modelo esquemático

Para realizar a aquisição de dados por meio dos sensores LDR, foi desenvolvido um circuito responsável por identificar a variação em cada sensor. Por meio de um divisor de tensão, utilizando o sensor LDR na parte inferior do divisor, e entre o resistor

e o LDR terá o valor que deve entrar para o conversor AD do Arduino. Na Figura 8 pode-se ver que foram utilizados os pinos de 1 a 4 do conector J3 representados no esquemático.

O circuito desenvolvido para acionamento das fitas de LED está demonstrado na Figura 9 que utilizou os pinos 5 a 8 do conector J3. O valor de PWM acionados nas portas dos Arduino irá gerar sinais dos pinos 5 a 8, de acordo com a porta, irá acionar um conjunto de fitas de LED em uma área específica. Ao acionar o PWM com seu valor máximo, o transistor irá permitir passagem da tensão máxima de 12V para as fitas de LED.

Figura 8 – Esquemático do circuito de aquisição por LDR



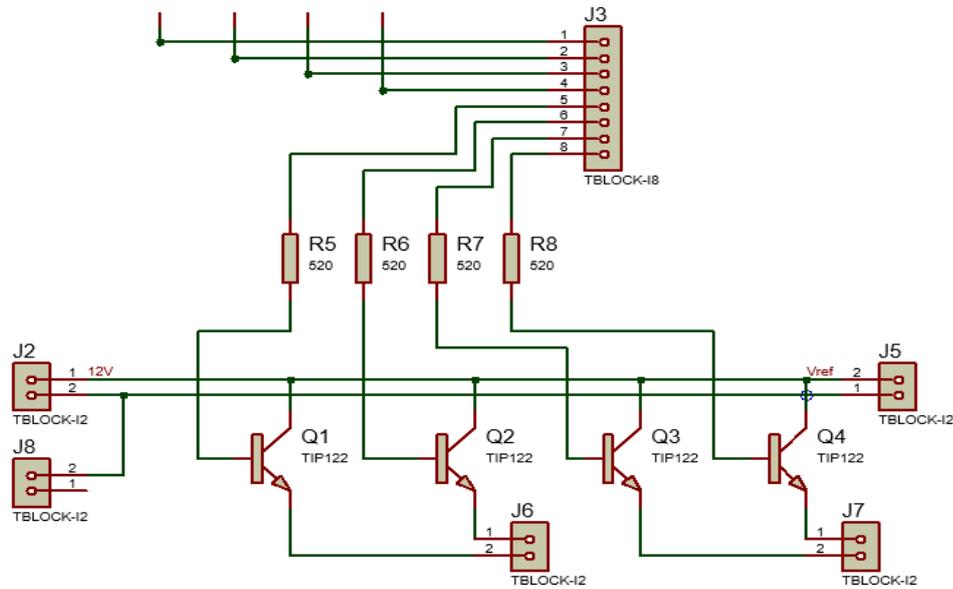
Fonte: Autor.

3.2.2 Arduino Mega

O Arduino Mega é um plataforma de prototipagem eletrônica que possui o microcontrolador ATmega2560, com 54 entradas e saídas já inclusas, que o tornam uma ferramenta flexível e de fácil utilização. Para o projeto em questão foram utilizadas as entradas analógicas, entradas e saídas digitais e saídas PWM. A Tabela 1 apresenta o detalhamento das portas utilizadas e suas conexões com a PCI.

Para esta solução, foi incluído em conexão com o Arduino um módulo de RTC - *Real Time Clock* (Relógio de Tempo Real) que permite informar a data e hora em alta precisão. Este módulo permite verificar a temperatura, porém não foi utilizado esse recurso no projeto. A Figura 10 mostra o aspecto das conexões de entrada e saída do Arduino com a PCI e o módulo de RTC.

Figura 9 – Esquemático do acionamento fitas das fitas de LED



Fonte: Autor.

Tabela 1 – Relação PCI e Arduino

Tipo	Fluxo	Porta	PCI	Descrição
Analógico	Entrada	0	S0	Sensor área 0
Analógico	Entrada	1	S1	Sensor área 1
Analógico	Entrada	2	S2	Sensor área 2
Analógico	Entrada	3	S3	Sensor área 3
Digital	Entrada	26	B0	Botão 0
Digital	Entrada	28	B1	Botão 1
Digital	Entrada	30	B2	Botão 2
Digital	Entrada	32	B3	Botão 3
Digital	Saída	27	D4	Led D4
Digital	Saída	29	D3	Led D3
Digital	Saída	31	D2	Led D2
Digital	Saída	33	D1	Led D1
Digital-PWM	Saída	2	Q4	Saída Fita LED Área 0
Digital-PWM	Saída	3	Q3	Saída Fita LED Área 1
Digital-PWM	Saída	4	Q2	Saída Fita LED Área 2
Digital-PWM	Saída	5	Q1	Saída Fita LED Área 3

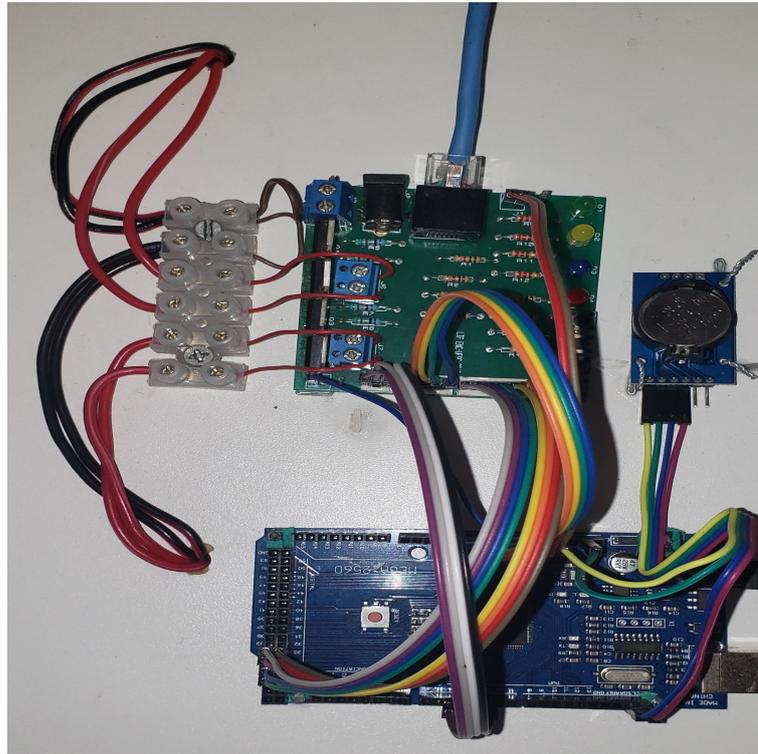
Fonte: Autor.

3.2.3 Materiais utilizados

Para o desenvolvimento desse trabalho foram utilizados os seguintes materiais:

- 1 Arduino MEGA com microcontrolador ATmega2650, com 54 pinos digitais, onde 14 são PWM. Além disso, possui 16 pinos analógicos e 16MHz de velocidade;
- 4 Sensores de Luminosidade LDR;

Figura 10 – PCI, Arduino e Módulo RTC



Fonte: Autor.

- 4 Transistores Bipolares do tipo NPN (*Negative-Positive-Negative*) TIP122;
- 4 Resistores 270R 5% (1/4W);
- 8 Resistores 1k 5% (1/4W);
- 4 Botões;
- 4 LEDs de cores diferentes;
- 2 Conectores 8 pinos;
- 2 Conectores 2 pinos;
- 3 Conectores Borne Azul 2 pinos;
- 1 Conector RJ45;
- 1 Conector fêmea para fonte de 2A;
- 1 Fita LED de 1m, Cor: Branca;
- 1 RTC DS3231;
- 1 Fonte de alimentação 12V 2A;
- 1 Luxímetro MLM-1020 Minipa;
- *Jumpers* (macho/macho e macho/fêmea).

3.3 Projeto de *Software*

Nesta seção será apresentado o projeto dos *softwares* desenvolvidos para execução deste trabalho.

3.3.1 Software Base

O desenvolvimento da solução em *software* que busca a integração correta e eficiente dos sistemas de iluminação natural e artificial, foi realizada por controle PI e programados pela IDE - *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado) do Arduino. Para analisar o problema proposto foi desenvolvido programas diferentes para cada cenário, com seus parâmetros e especificações que permitiu-se avaliar o funcionamento para os diferentes cenários anteriormente citados.

Quando se utiliza controle PI, é necessário utilizar apenas os valores de ganho proporcional e integral, definidos pelas constantes K_p e K_i . Ao implementar essa técnica na IDE do Arduino deve-se manter K_d zerado, determinando assim, que o sistema em questão é de fato PI.

Para utilizar a funcionalidade de controlador no Arduino, é necessário incluir uma biblioteca externa. A biblioteca está disponível no próprio gerenciador de biblioteca do Arduino, buscando por PID e realizando a instalação.

São necessárias as definições de variáveis como referência, entrada, saída, K_p , K_i e K_d que são do tipo *double*, decimal com precisão de 64 bit. Depois de definir uma referência e valores para o controlador, é necessário instanciar o controlador PID e passar o modo dele para automático no *setup* do Arduino.

A sintaxe da função PID é a seguinte: PID

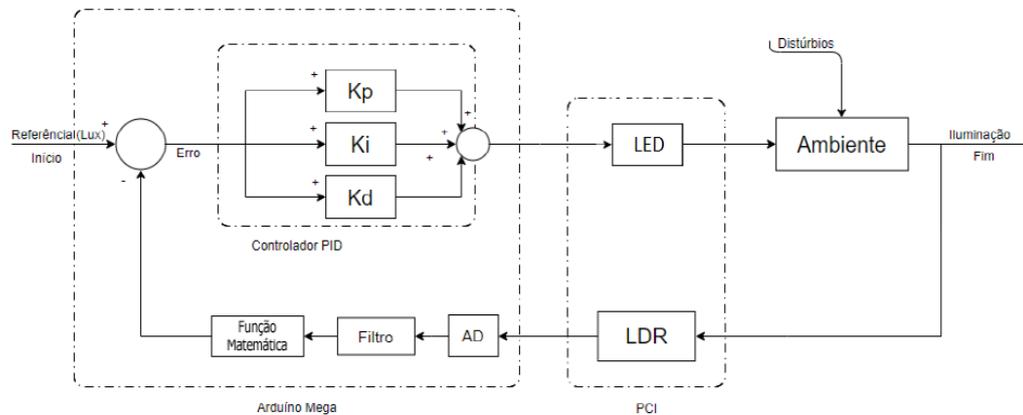
PID_Controller(entrada, saida, referencia, Kp, Ki, Kd, DIRECT);

Deve-se levar em consideração que ao trabalhar com valores que decrescem é necessário alterar o parâmetro para *REVERSE*, e ao trabalhar com valores que crescem deve-se utilizar parâmetro *DIRECT*. Neste trabalho, ao utilizar o valor de AD como entrada do controlador foi necessário alterar para *REVERSE*. Quanto maior for a luminosidade, menor é o valor de AD. O contrário acontece ao trabalhar com lx, logo se utiliza *DIRECT*.

O processo pode ser descrito através do Diagrama de Blocos na Figura 11. Foi considerado um valor de referência para a iluminância. A referência é comparada com o valor lido pelos sensores, gerando o sinal de erro. O controlador PI gera um sinal de controle que é aplicado no ambiente através do acionamento das fitas LED. Os sensores LDR identificam a intensidade e convertem para sinais analógicos de tensão. Esses sinais são lidos pelo Arduino, que realiza a conversão AD com 10 bits de resolução. O sinal passa por um filtro que será especificado para cada cenário.

Os *softwares* foram desenvolvidos apenas para funcionarem no modo de controle automático. Por esse motivo, ao energizar o sistema a placa Arduino já iniciará a execução de algum dos três cenários: uma área, duas áreas ou quatro áreas.

Figura 11 – Diagrama de blocos do processo de medição



Fonte: Autor.

3.3.2 Projeto dos Filtros

Para realizar a filtragem dos valores de entrada foram utilizados dois filtros: o exponencial e o filtro de passa-baixa de primeira ordem.

O filtro exponencial pode ser descrito pela equação (3.2), onde o coeficiente de suavização (α) pode variar entre $0 < \alpha < 1$. Assim, quanto menor o valor de α , mais amortecido será o valor aplicado no filtro. O Z_t representa o valor exponencialmente suavizado, e o somatório Z_0, Z_1, \dots, Z_n é uma série temporal estacionária.

$$Z_t = \alpha \sum_{k=0}^{t-1} (1 - \alpha)^k Z_{t-k} + (1 - \alpha)^t * Z_0, t = 1, \dots, N \quad (3.2)$$

Para exemplificar, temos a Equação (3.3) que representam a Equação (3.2) simplificada. Ao receber um valor novo, ele será multiplicado pelo coeficiente de suavização e somado a uma participação do valor antigo. Se utilizarmos α como 0.05, teremos só 5% do valor novo representando o ValorFinal. Nesse contexto, foi definido um valor de α para cada cenário proposto de forma experimental, buscando evitar efeitos estroboscópicos.

$$ValorFinal = \alpha * [ValorNovo] + (1 - \alpha) * [ValorAntigo] \quad (3.3)$$

A definição da posição do polo do filtro passa-baixa pode ser feita através da análise espectral dos sinais dos sensores. Assim, o polo será posicionado de forma a atenuar os sinais com frequências acima da frequência de corte. Para realizar a análise, deve-se primeiramente determinar o tempo de amostragem. Nesse trabalho, esse procedimento foi realizado utilizando o sistema em malha aberta, ou seja, é aplicado um valor fixo no PWM nas fitas LED. Enquanto isso, o programa apresenta na saída serial o período entre leituras, permitindo calcular o tempo de amostragem. Os dados

coletados são processados no Matlab gerando o gráfico da análise espectral. Na seção 4.1.1 do capítulo Resultados será abordado o processo de geração dos gráficos de cada cenário.

3.3.3 Mapeamento para conversão lx

Como parte da solução, os sensores LDR realizam a aquisição de dados. Esses dispositivos tem sua resistência alterada conforme a iluminância, o que muda a tensão nas entradas analógicas da placa Arduino. Os valores de tensão transformados de analógico para digital em uma faixa de 0 a 1023.

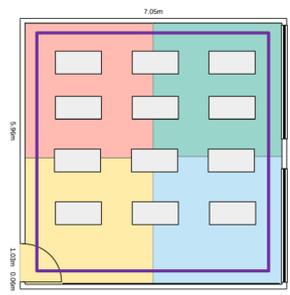
A proposta de haver uma função de conversão AD para lux é permitir a medição do nível de iluminação que o sensor está identificando. Considerando que o conversor AD tem 10 bits de resolução, possibilitando o uso de 0 a 1023 em Analógico para Digital (AD). Dentro dessa faixa, 1023 refere-se à um ambiente totalmente escuro e 0 refere-se a presença de alta iluminância.

Para estabelecer uma equação matemática, é necessário fazer um mapeamento da faixa de valores de iluminância e AD, permitindo que fosse possível parametrizar os valores, e possibilitando que pudessem ser correlacionados. O processo de medição será especificado na subseção 4.1.2.

3.3.4 Cenários Implementados

Nesta subseção serão apresentados os cenários implementados como parte da solução, busca-se tornar eficiente o uso da iluminação natural e artificial. Para todos os cenários, foi necessário encontrar o período de amostragem que será definido no próximo capítulo.

Figura 12 – Área representante de 1 área



Fonte: Autor.

3.3.4.1 Uma Área

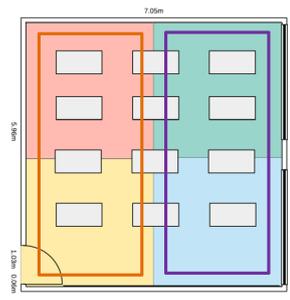
Para o desenvolvimento desta aplicação foi considerada uma área que contém as áreas A0, A1, A2 e A3 na Figura 12. Os valores de entrada dos quatro sensores

foram utilizados para calcular uma média e aplica-la no sistema de controle. Após realizar o processamento, o mesmo valor de saída é aplicado nas fitas de LED das quatro áreas (A0, A1, A2 e A4), aumentando ou diminuindo a iluminância artificial. Neste cenário foi utilizado apenas um controlador.

3.3.4.2 Duas Áreas

Nesse caso foram consideradas duas áreas: os valores de entrada dos sensores mais próximos da janela (A0 e A1) foram utilizados para calcular a média A01 e aplicá-la no controlador PI. A mesma ação foi realizada para os sensores mais distantes da janela (A2 e A3), que foram aplicados em outro controlador com as mesmas configurações. As duas áreas representadas na Figura 13. Após realizar o processamento, os valores de saída dos dois controladores foram aplicados nas fitas LED localizadas nas áreas A0, A1 e em A2 e A3.

Figura 13 – Área representante de 2 áreas

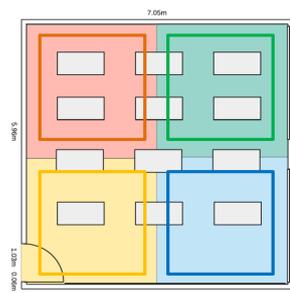


Fonte: Autor.

3.3.4.3 Quatro Áreas

Para o desenvolvimento desta aplicação foram consideradas quatro áreas na Figura 14. O valor de entrada de cada sensor foi aplicado em um controlador responsável pela área utilizando PI. Após realizar o processamento, os valores de saída dos controladores foram aplicados em cada área A0, A1, A2 e A3.

Figura 14 – Área representante de 4 áreas



Fonte: Autor.

3.3.5 *Softwares* Utilizados

Para realizar o desenvolvimento desse trabalho foram utilizados os seguintes *softwares*:

1. IDE Arduino V. 1.8.8
2. MegunoLink V. 1.25.18329.1125
3. MLM-1020 Interface Program V. 2.00
4. Microsoft Excel 2007
5. MatLab R2016a

4 RESULTADOS

Neste capítulo será apresentado o processo de testes e os resultados encontrados a partir do desenvolvimento deste trabalho.

4.1 Testes e Configuração

Nesta seção será apresentado todo o processo para a realização dos testes, parametrização dos filtros e desenvolvimento das funções matemáticas de conversão dos valores lidos pelos sensores.

Todo processo de medição realizado com o ambiente em escala foi realizado com orientação 106 graus à leste. Os testes foram realizados entre os dias 10/06/2019 e 19/06/2019 buscando um padrão de dia ensolarado, sem a presença de nuvens. Iniciando as medições no amanhecer das 06:30h até as 12:00h, porém entre uma e duas hora após o início já não necessário utilização de luz artificial.

4.1.1 Filtros

A aplicação de filtros serve para obter dados mais precisos, minimizando os efeitos dos ruídos presentes nas altas frequências. Para o trabalho em questão, não foi identificado a origem do ruído presente.

Para definir os filtros passa-baixa de primeira ordem foi necessário realizar alguns processos, incluindo a análise espectral de frequência que permite verificar onde existe maior incidência de ruídos. Para o filtro exponencial, foi definido um valor de α para cada cenário. Os resultados serão especificados nas subseções a seguir.

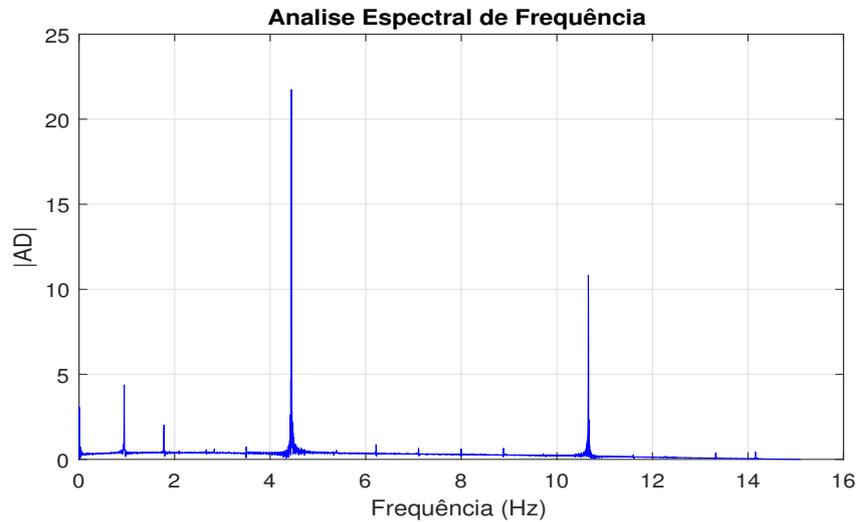
4.1.1.1 Uma Área

Neste cenário, para realizar a filtragem dos valores de entrada foram testados dois filtros: o exponencial com $\alpha = 0,10$ e um filtro de passa-baixa de primeira ordem.

Para determinar o filtro de passa-baixa foi executado o *software* em malha aberta e encontrado tempo de amostragem $t_s = 0,03311$ s. O código-fonte utilizado pode ser visto no Apêndice A. Assim, foi possível realizar a análise espectral de frequência do sinal dos sensores deste cenário. Esta análise auxilia na identificação das incidências de ruídos, que se projetado corretamente, realizará a filtragem desses dados deixando-os mais precisos.

Conforme pode-se ver na Figura 15 existem picos de ruído em 1 Hz. Ou seja, é necessário escolher uma frequência de corte menor que este valor. Assim, com a aplicação em testes o melhor valor entrado foi $f_c = 0,1$ Hz.

Figura 15 – Espectro de frequência - cenário uma área



Fonte: Autor.

A partir disso, é necessário verificar qual é a taxa de amostragem do sistema em malha fechada. Foi utilizado o código-fonte especificado no Apêndice D. Logo, tem-se $T_s = 0,110928$ s. Utilizando o Matlab com os valores de T_s e f_c é possível definir a função de transferência do filtro (H). Com a função de transferência e o T_s é possível encontrar a sua forma discreta (Hd). Para estabelecer a função de transferência é utilizada a função tf do Matlab na forma $H = tf(num, den)$, o numerador é definido como $num = 2 * \pi * f_c$ e o denominador $den = [1 \quad num]$. Para o caso em questão, as expressões encontradas são vistas nas equações (4.1) e (4.2)

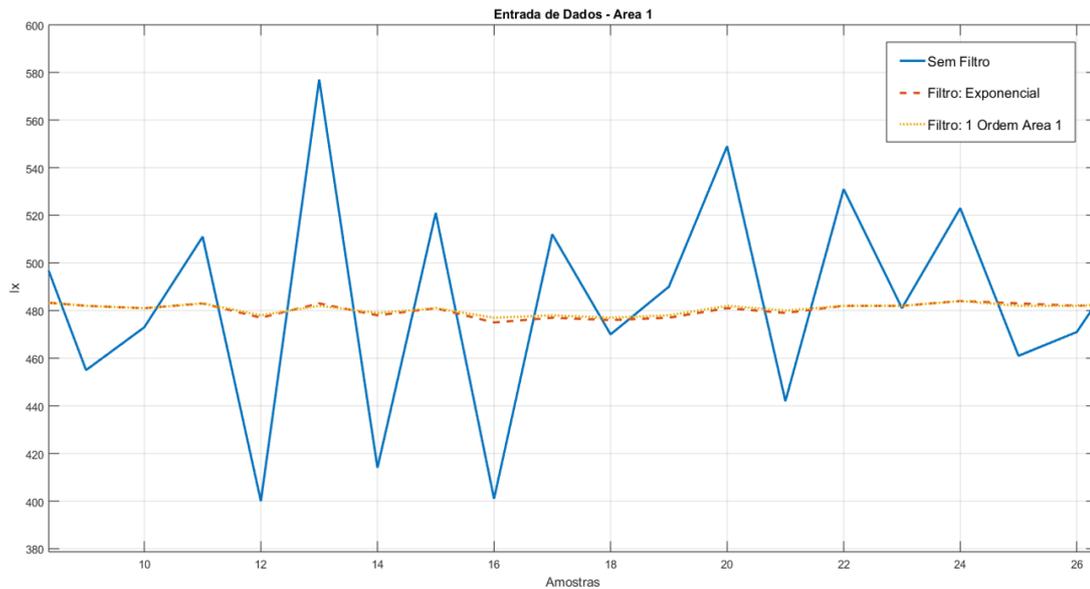
$$H = \frac{0,6283}{s + 0,6283} \quad (4.1)$$

$$Hd = \frac{0,06732}{z - 0,9327} \quad (4.2)$$

Para encontrar o Hd foi utilizada a função do Matlab $c2d$, cuja sintaxe é $Hd = c2d(H, T_s)$. O filtro aplicado no sinal analógico de entrada atenua os ruídos em altas frequências, tornando o valor mais adequado para atuação do controlador.

Neste contexto, temos a resposta em malha fechada dos filtros de primeira ordem e o exponencial demonstrados na Figura 16. Nota-se que de acordo com os valores definidos, os filtros se comportam similarmente. Por esse motivo, optou-se por realizar os testes de malha fechada utilizando apenas o filtro de primeira ordem.

Figura 16 – Entrada de dados com e sem filtro - 1 Área



Fonte: Autor.

4.1.1.2 Duas Áreas

Para este cenário foi realizada a execução do *software* em malha aberta, cujo código-fonte encontra-se no Apêndice B, encontrando tempo de amostragem $t_{s_{01}} = 0,077944$ s e $t_{s_{23}} = 0,077712$ s. Assim, foi possível realizar a análise espectral de frequência deste cenário. Conforme pode-se ver nas Figuras 18a e 18b, existem picos de ruído próximos de 1 Hz, Assim, foi necessário escolher uma frequência de corte menor que este valor, $f_c = 0,05$ Hz.

A partir disso, é necessário verificar qual é a taxa de amostragem do sistema em malha fechada, utilizando o código-fonte visto em Apêndice E. Logo, tem-se $T_{s_{01}} = 0,297376$ s e $T_{s_{23}} = 0,297144$ s. Nota-se pela Figura 18a e Figura 18b que os ruídos presentes estão nas mesmas frequências, e as taxas de amostragem se mantiveram próximas. Utilizando o Matlab com os valores de T_s e f_c é possível definir a função de transferência pela equação (4.3) e encontrar o equivalente discreto na equação (4.4), empregando as mesmas funções mencionadas no item anterior.

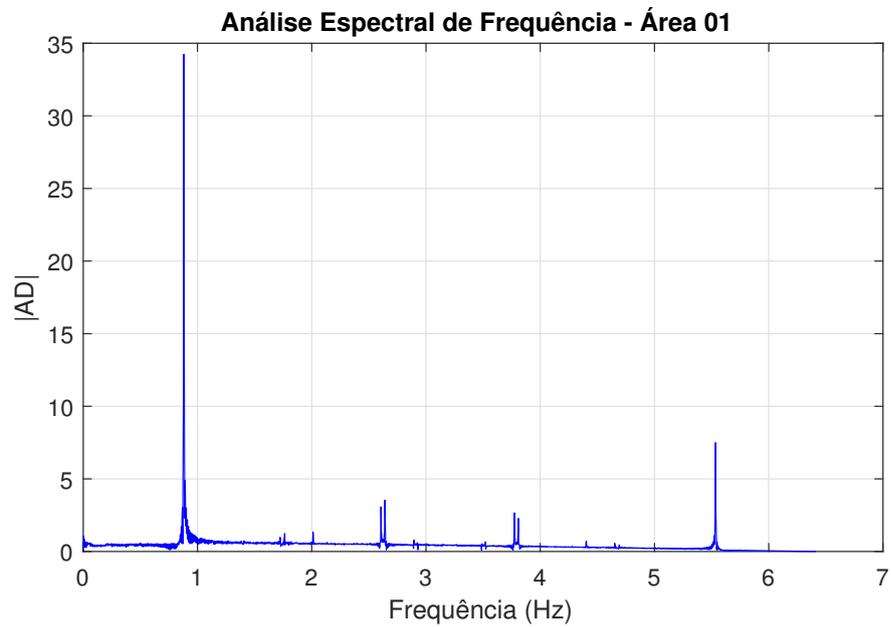
$$H = \frac{0,3142}{s + 0,3142} \quad (4.3)$$

$$Hd = \frac{0,08919}{z - 0,9108} \quad (4.4)$$

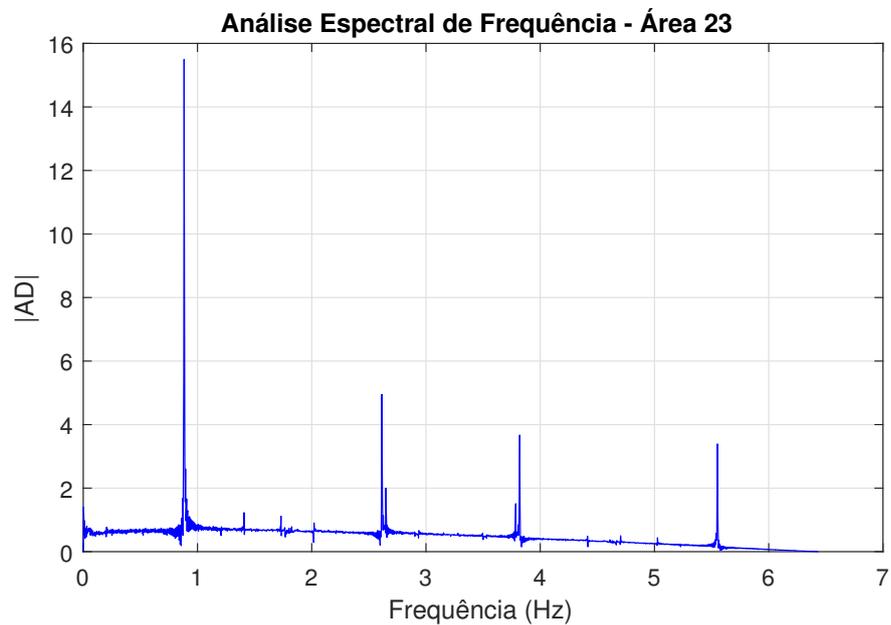
Neste cenário, para realizar a filtragem dos valores de entrada foram testados dois filtros: o exponencial com $\alpha = 0,05$ e o filtro de passa-baixa de primeira ordem, conforme especificado anteriormente, o filtro exponencial com alpha determinado atua

Figura 17 – Espectro de frequência - Cenário de duas áreas

(a) Espectro de Frequência Área 01



(b) Espectro de Frequência - Área 23

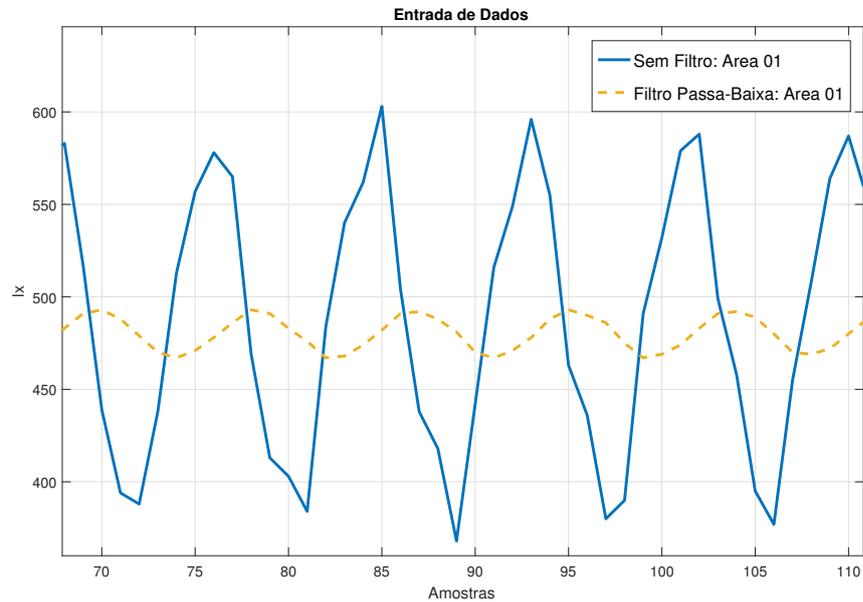


Fonte: Autor.

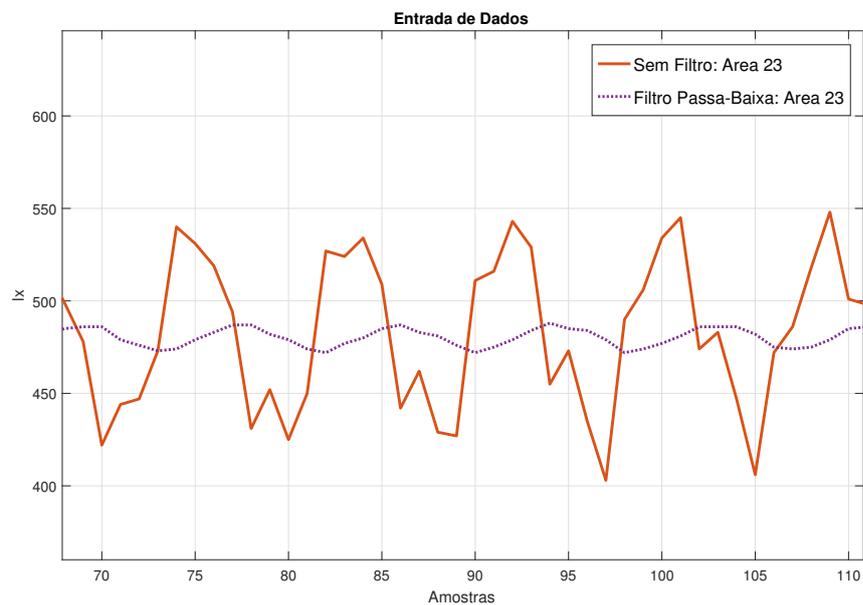
da mesma forma que o filtro de primeira ordem. Nota-se na Figura 18 que o filtro passa-baixa realiza uma boa filtragem de ruídos.

Figura 18 – Entrada de dados sem filtro e filtro passa-baixa 2 Áreas

(a) Cenário 2 áreas - Área 01



(b) Cenário 2 áreas - Área 23



Fonte: Autor.

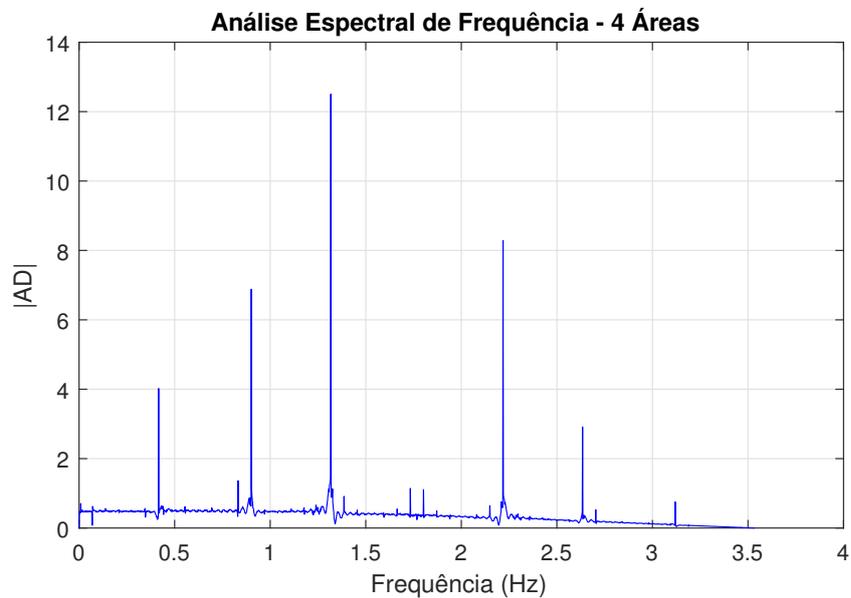
4.1.1.3 Quatro Áreas

Para este cenário foi realizada a execução do *software* em malha aberta, código-fonte em Apêndice C e foram encontrados quatro valores de tempo de amostragem. Porém, como os mesmos apresentaram valores muito próximos foi considerado $t_s =$

0,141312s. Com esses dados, foi possível realizar a análise espectral de frequência deste cenário. Conforme pode-se ver na Figura 19, existem picos de ruído próximos de 0,5 Hz, ou seja, foi necessário escolher uma frequência de corte menor que este valor, assim $f_c = 0,1$ Hz.

Para encontrar a taxa de amostragem do sistema em malha fechada foi utilizando o código-fonte em Apêndice F. Assim, foi possível encontrar a taxa de amostragem $T_s = 0,582400$ s. Empregando o Matlab com os valores de T_s e f_c , é possível definir a função de transferência conforme equação (4.6) e transformar a função de transferência para modelo discreto conforme a equação (4.6) utilizando as mesmas funções anteriormente mencionadas.

Figura 19 – Espectro Frequência de 4 Áreas



Fonte: Autor

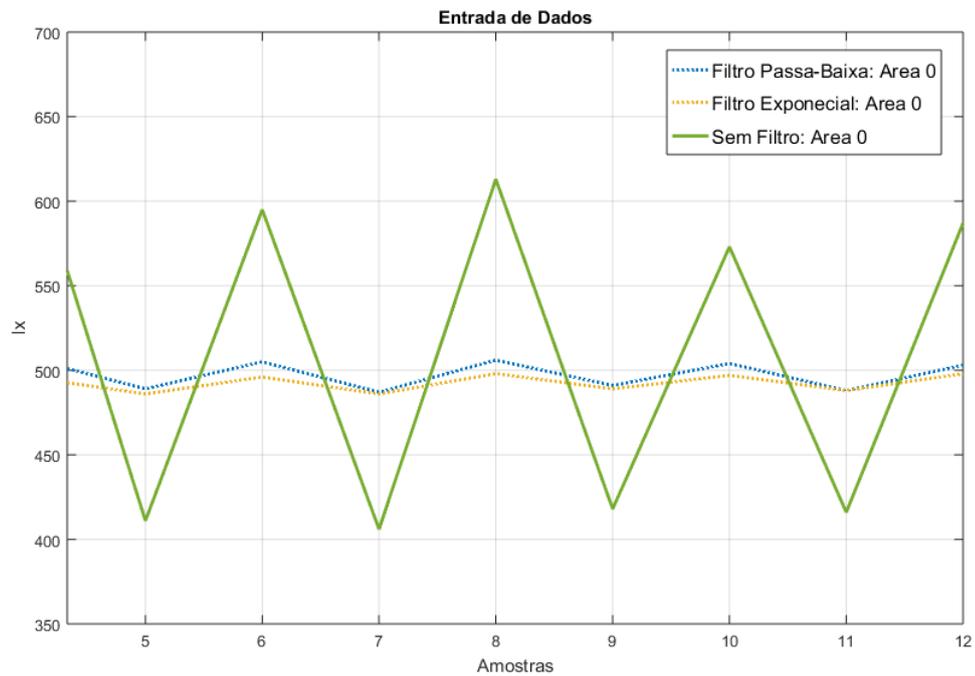
$$H = \frac{0,3142}{s + 0,3142} \quad (4.5)$$

$$Hd = \frac{0,1672}{z - 0,8328} \quad (4.6)$$

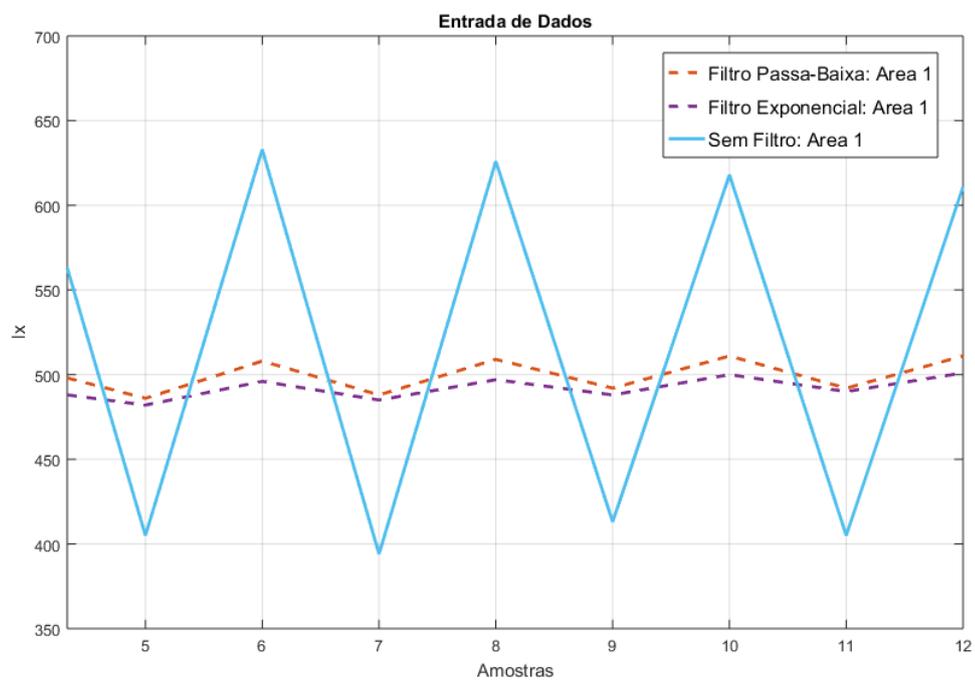
Na figura 20 temos o cenário de 4 áreas, onde está presente os valores dos sensores A0 e A1 respectivamente Figure 20a e Figura 20b. Tem-se os valores de entrada sem filtro, e os valores de entrada já filtrado com exponencial e passa-baixa. Nota-se que os valores estão bem próximos, como acontece com os demais cenários. Os filtros ajustados corretamente evitam os ruídos que prejudicariam o controle, vemos que ambos os filtros apresentarem resultados consideráveis.

Figura 20 – Entrada de dados sem filtro, filtro passa-baixa e filtro exponencial - 4 Áreas

(a) Cenário 4 áreas - Área 0



(b) Cenário 4 áreas - Área 1



Fonte: Autor.

4.1.2 Função Matemática de Conversão para lx

Para estabelecer uma equação matemática de conversão, é necessário fazer um mapeamento da faixa de valores de iluminância e AD conforme especificado na subseção 3.3.3. Mapeando todos os sensores, tem-se os dados apresetes na Tabela 2.

Nota-se que mesmo utilizando a potência máxima das fitas de LED não zera o valor de AD, ou seja, só irá diminuir mais o valor de AD quando a luminosidade for superior ao valor de lux definido como 100%. Ou seja, conforme a luminosidade da luz natural entrar em contato com o sensor e esse valor for superior, deverá diminuir ainda mais o valor de AD. Lembrando que a intensidade da luz natural sem contato com os raios de luz do sol pode chegar 25.000 lx e em contato com os raios, o valor pode chegar a 100.000 lx.

Tabela 2 – Dados dos Sensores A0, A1, A2, A3

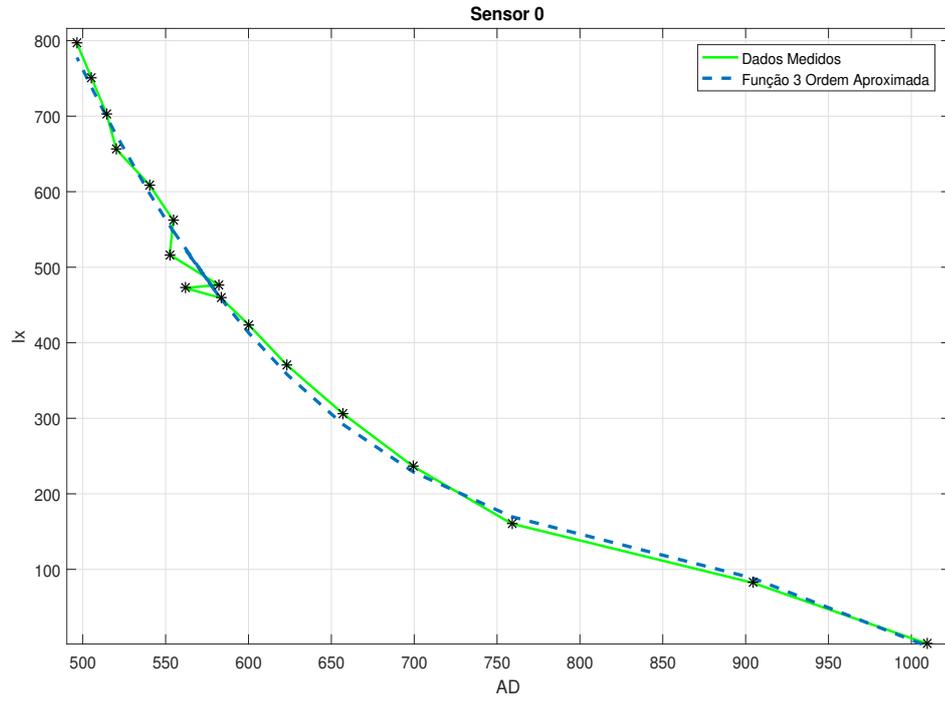
%	PWM	Lux0	AD0	Lux1	AD1	Lux2	AD2	Lux3	AD3
0	0	1,6	1009,5	1,4	1010,0	3,87	959,71	1,24	1010,4
6,25	16	82,2	904,3	68,4	860,8	81,02	841,77	92,12	770,6
12,50	32	160,4	758,9	149,9	749,8	154,5	715,99	179,6	680,0
18,70	48	236,4	699,3	230,6	685,8	227,4	641,05	269,8	627,2
25,00	64	305,6	657,3	306,0	656,8	293,0	591,34	355,0	583,2
31,25	80	369,8	623,5	374,8	629,6	352,4	556,94	431,6	550,6
37,50	96	423,4	600,3	434,0	607,6	406,2	533,24	499,8	532,0
43,75	112	459,4	583,3	482,2	592,6	443,8	517,25	558,0	514,8
50,00	128	462,4	561,7	509,2	582,0	457,0	504,58	563,4	505,8
56,25	144	476,6	581,9	518,2	578,6	467,6	504,37	605,4	502,6
62,50	160	516,6	552,5	533,4	572,2	492,6	495,53	605,6	502,4
68,75	176	562,2	554,6	577,4	562,2	543,2	481,37	649,0	492,4
75,00	192	609,0	540,2	624,8	547,8	578,0	474,29	698,2	480,0
81,25	208	656,0	520,3	673,2	536,2	622,4	465,78	753,4	467,8
87,50	224	703,6	514,3	721,6	525,8	667,0	451,86	807,6	457,2
93,75	240	751,0	505,2	760,2	517,6	711,2	442,68	860,6	446,8
100,0	255	796,4	496,5	817	510,8	753,2	433,59	912,0	439,0

Fonte: Autor

Analisando as figuras 21, 22, 23 e 24 em 500 lx, nota-se que para cada uma delas o valor de AD será diferente. Por esse motivo, foi necessário encontrar uma função matemática que representasse cada sensor. Foi empregada a função *polyfit* do Matlab, cuja sintaxe é $polinomio = polyfit(AD, LUX, N)$, onde N é a ordem da função a ser aproximada ao conjunto de dados.

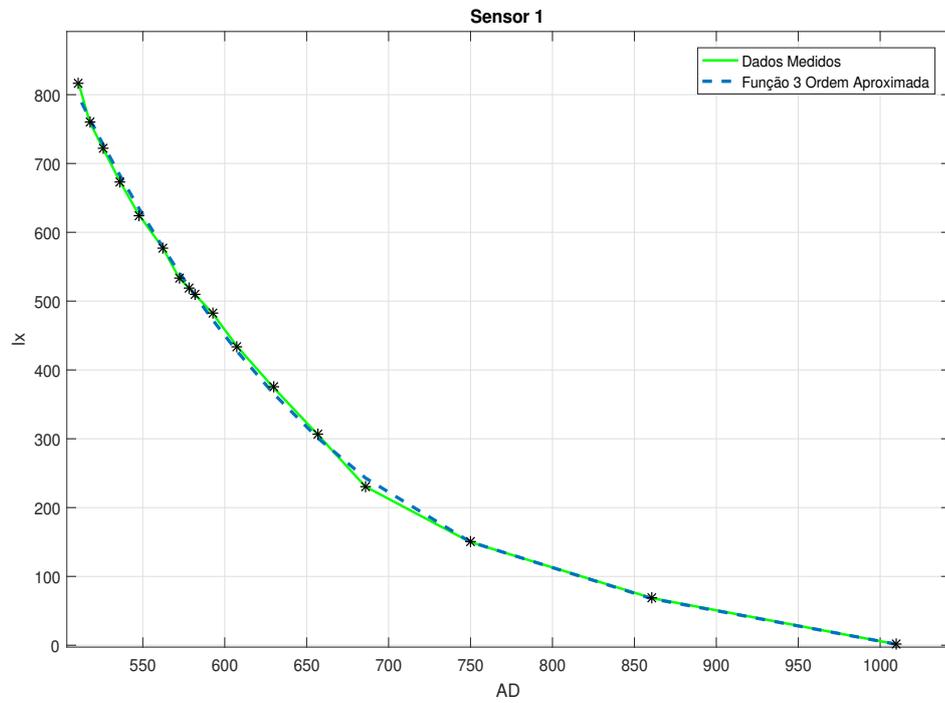
Para a escolha do melhor polinômio que represente cada sensor, foi analisada a faixa de valores de AD entre 0 a 1023 e as raízes do polinômio aproximado. Considerando que em todos os pontos da faixa de valores de AD não exista nenhuma

Figura 21 – Valores Medidos e Aproximação - Sensor 0



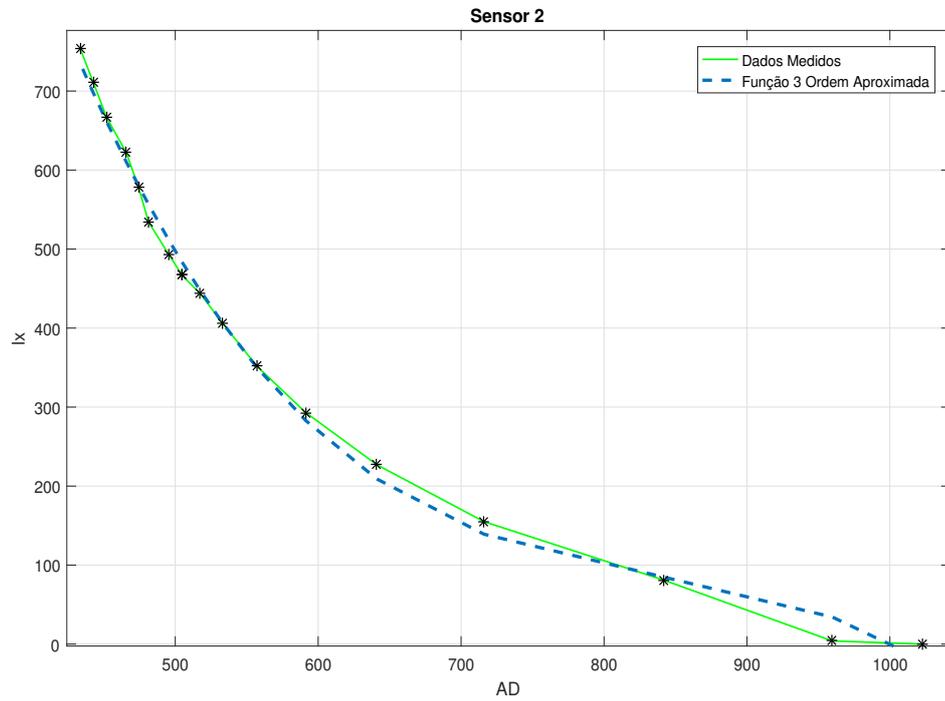
Fonte: Autor.

Figura 22 – Valores Medidos e Aproximação - Sensor 1



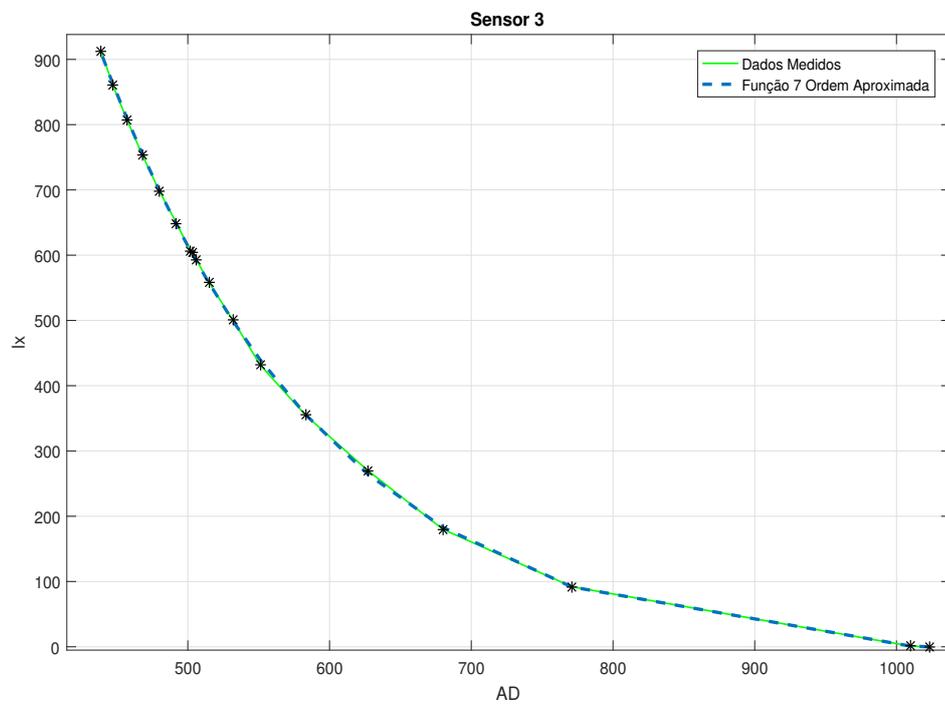
Fonte: Autor.

Figura 23 – Valores Medidos e Aproximação - Sensor 2



Fonte: Autor.

Figura 24 – Valores Medidos e Aproximação - Sensor 3



Fonte: Autor.

raiz, dessa forma, a curva de medição aproximando-se da curva gerada pelo polinômio encontrado. Assim, as figuras 21, 22, 23 e 24 mostram a relação entre os valores medidos e a aproximação da função. De acordo com o parâmetro das raízes foi escolhido ordem 3 para os sensores A0, A1 e A2 e ordem 7 para A3 conforme mostra as equações (4.7), (4.8), (4.13) e (4.10) abaixo.

$$f_0(x) = -1,01 * 10^{-5} * x^3 + 2,63 * 10^{-2} * x^2 - 2,31 * 10 * x + 7,03 * 10^3 \quad (4.7)$$

$$f_1(x) = -9,22 * 10^{-6} * x^3 + 2,71 * 10^{-2} * x^2 - 2,75 * 10 * x + 9,61 * 10^3 \quad (4.8)$$

$$f_2(x) = -7,49 * 10^{-6} * x^3 + 2,03 * 10^{-2} * x^2 - 1,84 * 10 * x + 5,82 * 10^3 \quad (4.9)$$

$$f_3(x) = -2,76 * 10^{-17} * x^7 + 1,45 * 10^{-13} * x^6 - 3,34 * 10^{-10} * x^5 + 4,47 * 10^7 * x^4 - 3,86 * 10^{-4} * x^3 + 2,20 * 10^{-1} x^2 - 7,80 * 10 * x + 1,32 * 10^4 \quad (4.10)$$

4.1.3 Controlador PI

Para o desenvolvimento desse trabalho foi utilizado controlador PI, sendo definido como valor de referência 480 lx. Para sintonizar o controlador, foram realizados testes nos sistemas em malha fechada, buscando evitar oscilações no valor de controle aplicado nas fitas LED. Assim, foram determinados os seguintes parâmetros.

$$Kp = 0,075 \quad (4.11)$$

$$Ki = 0,30 \quad (4.12)$$

$$Kd = 0 \quad (4.13)$$

4.1.4 Potência das fitas de LED

Inicialmente, foi realizada a medição da corrente das 4 fitas de LED presentes no ambiente em escala. Dessa forma, foi possível calcular a potência utilizando a Equação (4.14). Os valores de potência para cada fita estão presentes na Tabela 3.

Tabela 3 – Tensão, Corrente e Potência das áreas A0, A1, A2, A3 - Potência Máxima

Área	Tensão(V)	Corrente(mA)	Potência(W)
A0	12	24,4	0,2916
A1	12	26,2	0,3144
A2	12	25,1	0,3012
A3	12	27,3	0,3276

Fonte: Autor.

Assim, foi possível encontrar a potência máxima de acionamento das fitas de LED em sua utilização máxima de 1,2248 W.

$$P = V * I \quad (4.14)$$

Para um cenário de ajuste manual da luminosidade com a utilização de um dimerizador, levou-se em consideração a intensidade da luz no ponto central do ambiente sendo em torno 480 lx. Para atingir essa iluminância foi ajustado o PWM para 96, o que representa 37,5% do seu uso conforme especifica a Tabela 2. Nessa condição foram determinados os seguintes valores de potências, vistos na Tabela 4.

Ao acionar uma lâmpada tradicionais ela utiliza sua potência máxima quando está ligado, ou seja, não realiza nenhum tipo de ajuste de intensidade. As lâmpadas tradicionais permanecem em dois estados: ligadas, quando utilizam o máximo de potência, e desligadas quando a potência é zero.

Tabela 4 – Tensão, Corrente e Potência das áreas A0, A1, A2, A3 - *Dimmer*

Área	Tensão(V)	Corrente(mA)	Potência(W)
A0	12	14,2	0,1704
A1	12	14,9	0,1788
A2	12	14,8	0,1776
A3	12	15,5	0,1860

Fonte: Autor.

Quando compara-se a potência máxima de uso das lâmpadas LED com a utilização do *dimmer* (Figura 25), se obtém uma diferença de 41,8% de potência elétrica.

4.2 Resultados em Malha Fechada

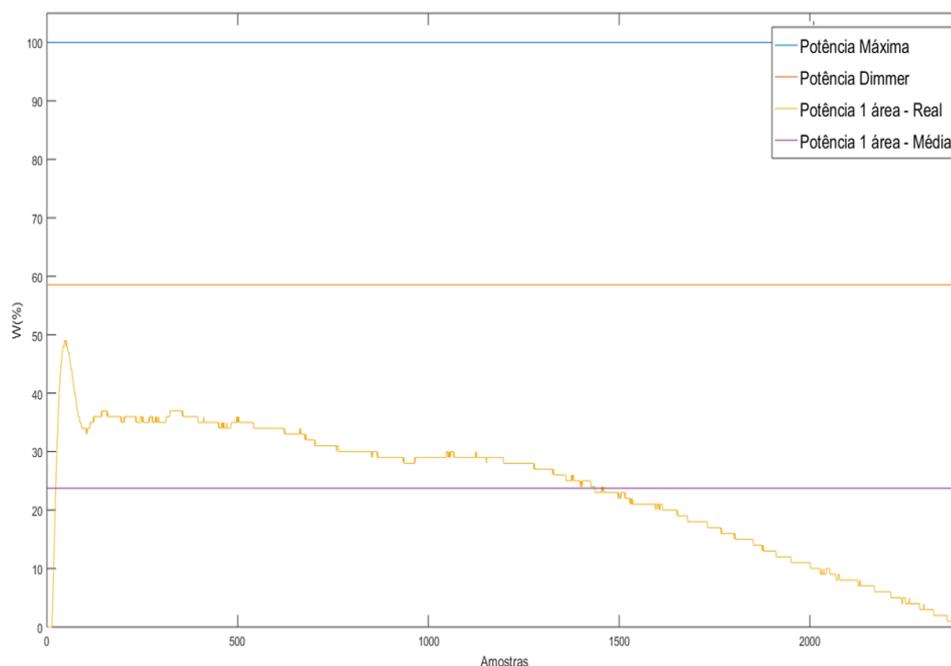
Todo processo de medição com o ambiente em escala foi realizado com orientação 106 graus à leste, durante o amanhecer entre os dias 10/06/2019 e

19/06/2019. Cada cenário teve a necessidade de ser testado em dias diferentes, porém buscou-se realiza-los em dias ensolarados e sem muitas nuvens.

Para o cenário de uma área, inicialmente, podemos comparar a utilização máxima de potência das lâmpadas LED com a potência utilizada pelo *dimmer*, e nota-se que há uma redução de potência elétrica que equivale a 41,8%. De acordo com a medição em tempo real, o controlador PI foi ajustando o uso das fitas de LED o que trouxe uma variação da utilização de potência elétrica das fitas, demonstrado pela Potência 1 área - Real (Figura 25).

Dentro dessa período foi realizado a média de uso do controlador e demonstrado na figura como Potência 1 área - média. Esse valor representou 24,23% em potência utilizada, ou seja, 0,2968 W. Assim, comparando potência média do controlador PI com a potência do *dimmer* houve uma redução de 33,97%. Ao realizar a comparação com a potência máxima, houve uma redução de 75,76% de potência utilizada para acionamento das fitas de LED com o controlador PI.

Figura 25 – Cenários 1 Área - Comparação de utilização em porcentagem



Fonte: Autor.

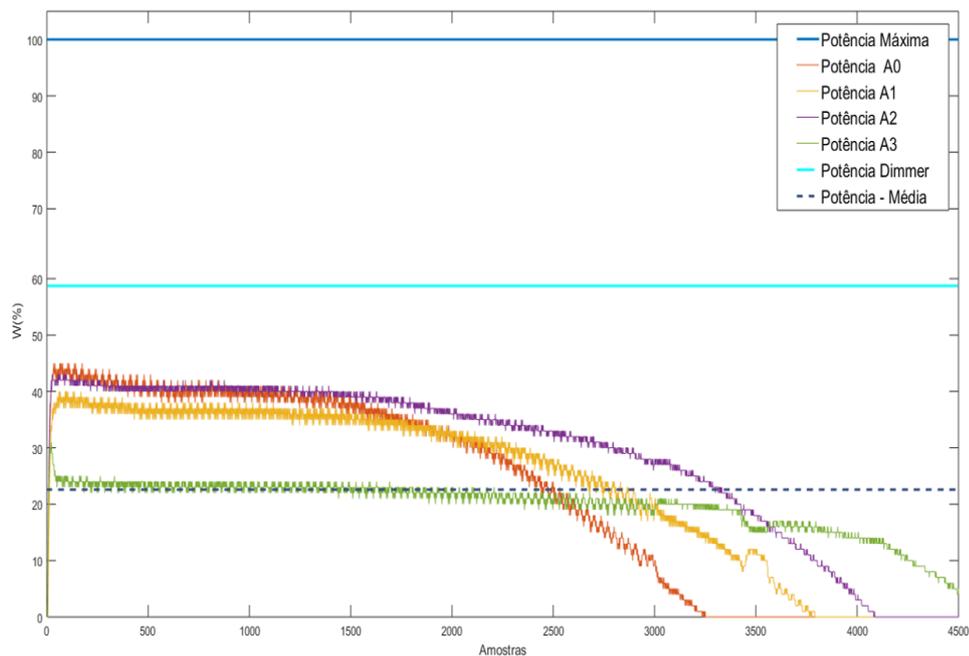
Para o cenário de quatro áreas, foi realizado o mesmo processo de comparação, potência máxima das fitas de LED, potência *dimmer*, potência real de cada controlador e potência média entre os 4 controladores. Assim, dentro dessa período foi realizado a média de uso dos controladores PI e demonstrado na figura 26) como Potência - média. Esse valor representou 21,35% em potência utilizada, ou seja, 0,2615 W.

Na área A0 foi utilizado 18,91% da potência total para acionamento das fitas de

LED desta área, na área A1 de 21,09%, área A2 26,13% e na área A3 18,72%. Essas diferenças podem ser explicadas devido a diferença de potência entre cada uma delas, demonstrados anteriormente na Tabela 3.

Comparando a média de potência utilizada pelos controlares com a potência de *dimmer*, tem-se uma redução de 36,85%. Quando comprado a média de potência utilizada com a potência máxima tem-se uma redução de 78,65%.

Figura 26 – Cenários 4 Áreas - Comparação de utilização em porcentagem



Fonte: Autor.

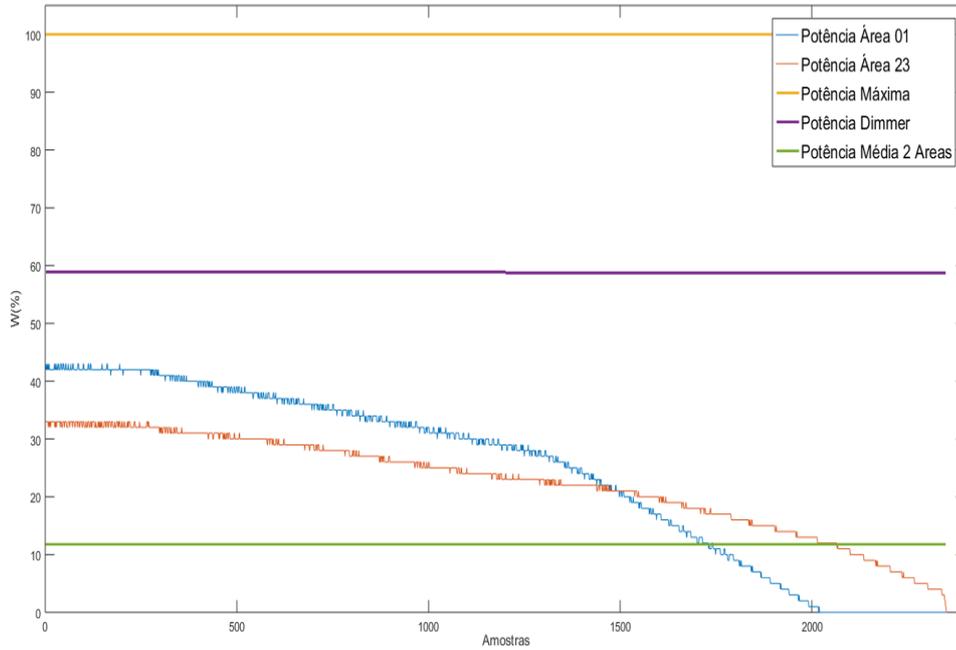
Para o cenário de duas áreas foram realizados dois testes, um deles mantendo o padrão dos anteriores, e outro ocultando com uma folha branca os espaços dentro do ambiente em escala que representam a porta e o quadro-negro (Figura 6c).

Para o cenário de duas áreas sem ocultar o quadro-negro e a porta, temos a resposta de porcentagem de potência para cada área A01 e A23 na Figura 27. A potência média das duas áreas, que foi utilizada para acionamento das fitas de LED foi de 0,1443W, representou 11,78% da potência total. Dessa forma, tem-se redução de 88,43% de uso de potência elétrica ao utilizar um controlador PI comparando com a potência máxima.

O cenário ocultando o quadro-negro e a porta com uma folha branca, tem-se a potência utilizada de 0,1172W, que representou 9,57% do uso de potência conforme mostra na Figura 28. Essa diferença de 2,21% entre os dois cenários pode ser explicada devido ao fenômeno de reflexão: as cores escuras tendem a absorver a luminosidade, enquanto as cores claras a refletir. Assim, quanto mais claro o ambiente for projetado,

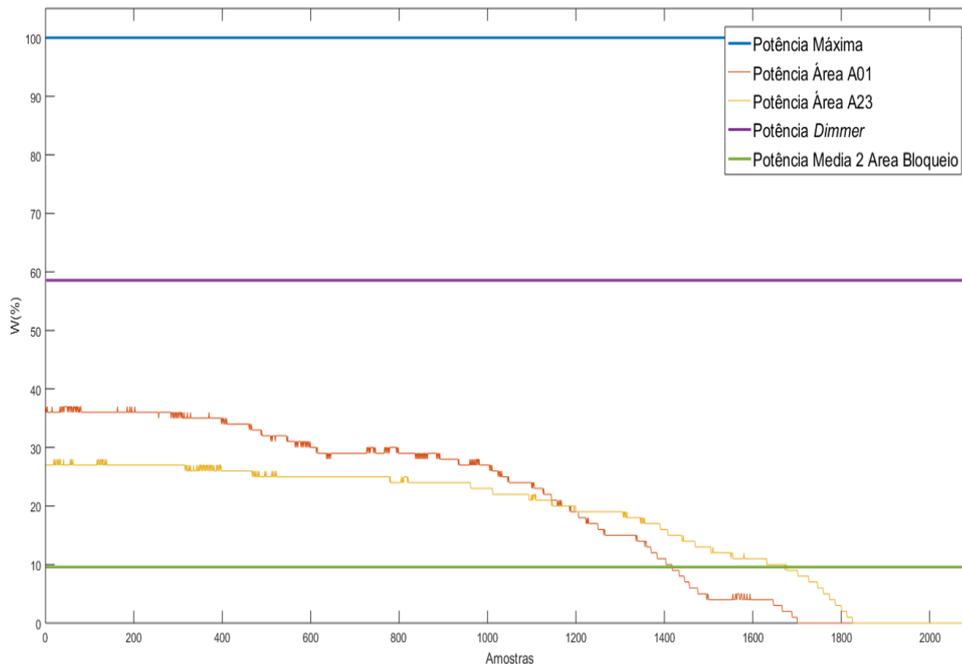
menos energia elétrica será necessária para mantê-lo com o valor de parâmetro desejado.

Figura 27 – Cenários 2 Áreas - Comparação de utilização em porcentagem



Fonte: Autor.

Figura 28 – Cenários 2 Áreas - Bloqueio da porta e quadro-negro



Fonte: Autor.

5 CONSIDERAÇÕES FINAIS

Esse trabalho apresentou a implementação de sistemas de controle buscando analisar as melhores técnicas para realizar a utilização eficiente da iluminação natural e artificial, buscando reduzir o consumo de energia.

Quando se trata de iluminação alguns parâmetros devem ser levados em consideração, tais como: nível de iluminância, de acordo com padrões de iluminância para ambientes, contrastes, ajuste de brilho e ofuscamento para melhorar as condições de iluminação do ambiente. Deve-se levar em conta que alto nível de iluminância não significa boas condições de iluminação e eficiência.

Arquitetos e engenheiros desenvolveram durante anos diversas técnicas para tornar a iluminação dos ambientes mais eficiente, não apenas na construção e planejamento de um edifício novo mas também em edifícios já construídos, implementando sistemas de controle e utilizando equipamentos adequados a essas técnicas.

A implementação de uma solução básica como a utilização de *dimmers* em ambientes pode reduzir o uso de energia elétrica. Um ponto negativo é a necessidade de ajuste manual, sem a utilização de sensores ou aparelhos de medição. Os usuários podem deixar em um nível maior de iluminância que o necessário, o que acaba deixando de ser tão eficiente energeticamente quanto poderia ser.

Quando utiliza-se um sistema automático que ajusta a luminosidade de acordo com um valor padrão, busca-se otimizar e consumir o mínimo de energia possível, aproveitando ainda a iluminação natural. Essa prática trará diferença de luminosidade quando comparado com um sistema tradicional de iluminação (ligado/desligado), já que este fará o uso total de sua potência, e conseqüentemente gastará mais energia.

Adicionalmente, sistemas de controle apresentam diversas soluções com questões de iluminação. No presente trabalho a utilização do controlador PI trouxe resultados que comprovam a redução do uso de potência. Nesse contexto, pode-se mencionar que cada cenário executado trouxe uma economia.

O cenário de uma área representou redução de uso de potência em 75,76% da potência total. Com quatro áreas representou redução de 78,65% e duas áreas 88,22% de redução de uso de potência. Onde foi utilizando a luz artificial apenas em momentos necessários e com isso fazendo um melhor aproveitamento da luz natural, o que culmina em economia de energia.

Conforme mostram os resultados, o ambiente em escala que utilizou menos

energia para controlar a iluminação foi o cenário com duas áreas, onde um controlador regula a iluminância em uma área mais próxima da janela e outro controlador na área mais afastada. O ambiente mais próximo responde diretamente as interferências externas e o outro realiza um balanço entre o uso de energia elétrica. Em termos de economia de consumo, foi verificado experimentalmente que a utilização dessas técnicas trouxeram economia na utilização de energia elétrica. Esse resultado pode ser aplicável à uma sala em escala real, porém pode sofrer alterações devido ao seu tamanho e número de sensores.

Dentro de um ambiente em escala real pode-se utilizar mais de uma solução. Conforme já exposto, a utilização de sensores de presença aliados a sistemas de controle de iluminação dimerizável apresenta potencial para alcançar maior economia de consumo de energia elétrica.

5.1 Trabalhos Futuros

Existe uma ampla utilização de sistemas de controle na efficientização do consumo de energia elétrica. No presente trabalho foi estudada a implementação de três cenários em um ambiente em escala por áreas de sensores. Porém, existe a possibilidade de expandir as aplicações. Alguns dos possíveis trabalhos a serem desenvolvidos no futuro com o mesmo ambiente em escala incluem:

1. Melhorias no protótipo
 - Aprimorar a conversão entre AD e lx;
 - Utilizar rede de sensores sem fio;
 - Utilizar IoT - *Internet of Things* (Internet das Coisas) para estabelecer referências, controle e medição de valores.
2. Complementar a atuação do sistema de controle
 - Implementação de sistema de controle de brise nas janelas para evitar excesso de iluminação natural.

REFERÊNCIAS

ABNT. NBR 5413: Iluminância de interiores. p. 1–13, 1992.

ABNT. NBR 15215-4: Iluminação natural – Parte 4: Verificação experimental das condições de iluminação interna de edificações – Método de medição. p. 1–13, 2004.

ABNT. NBR ISO/CIE 8995-1: Iluminação de ambientes de trabalho. **Associação Brasileira de Normas Técnicas**, p. 46, 2013.

Begemann, S.H.A. et al. Daylight, artificial light and people. 1994.

BEZERRA, J. H.

Estudo de Sistema de Controle Usando o Controlador PID em Processos Industriais — Faculdade de Engenharia Mecânica, UNICAMP, 2007. Disponível em: <http://repositorio.unicamp.br/bitstream/REPOSIP/330913/1/Bezerra{_}JoseHenrique{_}.>

BILÉSIMO, T. L. Análise e efficientização do consumo de energia em um campus universitário. 2015.

BYUN, J.; SHIN, T. Design and Implementation of an Energy-Saving Lighting Control System Considering User Satisfaction. **IEEE Transactions on Consumer Electronics**, v. 64, n. 1, p. 61–68, 2018. ISSN 00983063.

Comissão Europeia. DIRECTIVA 2002/91/CE do Parlamento Europeu e do Conselho de 16 de Dezembro de 2002 relativa ao desempenho energético dos edifícios. **Jornal Oficial das Comunidades Europeias**, L 1, n. 11, p. 65–71, 2003.

De Carli, M.; De Giuli, V.; ZECCHIN, R. Review on visual comfort in office buildings and influence of daylight in productivity. **Indoor Air**, n. August, p. 17–22, 2008.

DIDONÉ, E. L.; OSCAR, F.; PEREIRA, R. O potencial da luz natural na economia de energia elétrica para a iluminação artificial. **Revista Tecnológica**, p. 24–34, 2009. Disponível em: <<http://periodicos.uem.br/ojs/index.php/RevTecnol/article/download/8708/5168.>>

FONSECA, R. W. da. A Influencia das Reflexoes Internas como Contribuição da Iluminação Natural no Ambiente Construído. p. 188, 2007.

GANDRA, A. **Venda de Lâmpadas Incandescentes está Proibida no País a Partir de Hoje**. 2016. Disponível em: <<http://agenciabrasil.ebc.com.br/geral/noticia/2016-06/inmetro-inicia-fiscalizacao-no-varejo-de-lampadas-incandescentes-de-41-w-60-}>>

GUL, M. S.; PATIDAR, S. Understanding the energy consumption and occupancy of a multi-purpose academic building. **Energy and Buildings**, Elsevier B.V., v. 87, p. 155–165, 2015. ISSN 03787788. Disponível em: <<http://dx.doi.org/10.1016/j.enbuild.2014.11.027.>>

IEA. **Energy Efficiency: Buildings**. 2018. Disponível em: <<https://www.iea.org/topics/energyefficiency/buildings/>>.

IEA. **Energy Efficiency: Lighting**. 2018. Disponível em: <<https://www.iea.org/topics/energyefficiency/buildings/lighting/>>.

IEA. **World Energy Outlook 2018**. 2018. Disponível em: <<https://www.iea.org/weo2018/electricity/>>.

LAMBERTS, R.; DUTRA, L.; PEREIRA, O. F. **A Eficiência Energética na Arquitetura**. 3. ed. [S.l.: s.n.], 2014.

MINISTÉRIO DO MEIO AMBIENTE (MMA). **Eficiência energética: guia para etiquetagem de edifícios**. [S.l.: s.n.], 2015. v. 2. 70 p. ISBN 0029-2001 (Print)\r0029-2001.

NIERO, J. **Tecnologia é aliada na redução do consumo de energia em residências**. 2017. Disponível em: <<http://www.fecomercio.com.br/noticia/tecnologia-e-aliada-na-reducao-do-consumo-de-energia-em-residencias>>.

P. Palensky, S. M. Latest Trends in Building Automation and Control System. n. March 2007, 2003.

PEREIRA, F. O. R.; PEREIRA, R. C.; CASTAÑO, A. G. Quão confiáveis podem ser os modelos físicos em escala reduzida para avaliar a iluminação natural em edifícios? **Ambiente Construído**, v. 12, n. 1, p. 131–147, 2017.

PÉREZ-LOMBARD, L.; ORTIZ, J.; POUT, C. A review on buildings energy consumption information. **Energy and Buildings**, v. 40, n. 3, p. 394–398, 2008. ISSN 03787788.

PETRY, B. M. et al. Potencial de Economia de Energia Elétrica através da Utilização de Sensores de Presença - Aplicação no Prédio 50 da PUCRS. **X Salão de Iniciação Científica - PUCRS**, p. 3036–3038, 2009.

PROCEL. Manual de iluminação eficiente. **Eletrobrás/Procel**, v. 1, p. 36, 2002.

PROCEL. Manual de Iluminação. p. 409–13, 2011.

ROBBINS, C. L. Daylighting, design and analysis. **New York, An Nostrand Reinhold Company**, p. 325, 1986.

SILVA, L. M. M. Iluminação no ambiente de trabalho visando o conforto ambiental. v. 01, p. 1–15, 2016.

SILVA, M. G.; TASCHETTO, R. L. Projeto De Um Sistema Automático De Controle De Luz Artificial Com Base Na Iluminação Natural Para a Redução Do Consumo De Energia. 2014.

SILVA, N. D. R. **Análise e Integração Eficiente da Iluminação Natural e Artificial em Uma Sala de Aula**. 2017. Disponível em: <<https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/178025/TCC{ }NATALIA.pdf?sequence=1>>.

SOUZA, R. V. G. Luz natural no projeto arquitetônico. Ed.31, p. 72–77, 2008.

TARIQ, W. et al. Building Management System for IQRA University. **Asian Journal of Engineering, Sciences & Technology**, v. 2, n. 2, p. 106–109, 2012.

WENDEN, A. L. Características de Sistemas de Iluminação Artificial. **MME**, v. 3, n. September, 2013.

YANG, S. H.; CHANG, Q. S.; CHOU, D. C. Empirical research of using digital addressing LED energy-saving lighting control system - The case of underground parking lot: Preparation of papers in two-column format for ISNE 2016. **2016 5th International Symposium on Next-Generation Electronics, ISNE 2016**, p. 2–3, 2016.

APÊNDICE A

```
1 //DEV_TCC_UFSC_ENC
2 //LEONARDO T FRANCIS
3 //Teste em Malha Aberta 1 Area
4 #include <Math.h>           //Para realizacao de calculos
5 #include <SoftwareSerial.h> //Para utilizar o Meguno
6 #include <MegunoLink.h>    //Para utilizar o Meguno
7
8 TimePlot   MyPlot;
9
10 // Portas PWM
11 const int pwmPin2 = 2;
12 const int pwmPin3 = 3;
13 const int pwmPin4 = 4;
14 const int pwmPin5 = 5;
15
16 // Portas Analogicas Sensores
17 const int li_sensorPin0 = 0;
18 const int li_sensorPin1 = 1;
19 const int li_sensorPin2 = 2;
20 const int li_sensorPin3 = 3;
21
22 // Variaveis Sensores e Lux
23 double ld_sensor0_old, ld_sensor1_old, ld_sensor2_old, ld_sensor3_old;
24 double ld_lux_area0123_sf;
25
26 void setup() {
27     Serial.begin(9600); //Saida Output Serial
28 }
29
30 void loop() {
31     f_start();          // Start
32 }
33
34 void f_start()
35 {
36     analogWrite(pwmPin2, 125);
37     analogWrite(pwmPin3, 125);
38     analogWrite(pwmPin4, 125);
39     analogWrite(pwmPin5, 125);
```

```
40
41  if (ll_s0 == 0)
42  {
43      tempo_inicio0 = micros();
44      ld_sensor0_old = analogRead(li_sensorPin0);
45      ll_s0 = 1;
46  }
47  else
48  {
49      tempo_fim0 = micros();
50      ll_s0 = 0 ;
51  }
52
53  if (ll_s1 == 0)
54  {
55      tempo_inicio1 = micros();
56      ld_sensor1_old = analogRead(li_sensorPin1);
57      ll_s1 = 1;
58  }
59  else
60  {
61      tempo_fim1 = micros();
62      ll_s1 = 0 ;
63  }
64
65  if (ll_s2 == 0)
66  {
67      tempo_inicio2 = micros();
68      ld_sensor2_old = analogRead(li_sensorPin2);
69      ll_s2 = 1;
70  }
71  else
72  {
73      tempo_fim2 = micros();
74      ll_s2 = 0 ;
75  }
76
77  if (ll_s3 == 0)
78  {
79      tempo_inicio3 = micros();
80      ld_sensor3_old = analogRead(li_sensorPin3);
81      ll_s3 = 1;
82  }
83  else
84  {
85      tempo_fim3 = micros();
86      ll_s3 = 0;
```

```
87     }
88
89     if ( ll_s0 == 0 )
90     {
91         Serial.print("A0_Time: ");
92         Serial.print(tempo_fim0    tempo_inicio0);
93     }
94     if ( ll_s1 == 0 )
95     {
96         Serial.print("A1_Time: ");
97         Serial.print(tempo_fim1    tempo_inicio1);
98     }
99     if ( ll_s2 == 0 )
100    {
101        Serial.print("A2_Time: ");
102        Serial.print(tempo_fim2    tempo_inicio2);
103    }
104    if ( ll_s3 == 0 )
105    {
106        Serial.print("A3_Time: ");
107        Serial.print(tempo_fim3    tempo_inicio3);
108    }
109
110    Id_lux_area0123_sf = (Id_sensor0_old + Id_sensor1_old + Id_sensor2_old
111                        + Id_sensor3_old)/4;
112
113    MyPlot.SendData("AD_sf", Id_lux_area0123_sf);
114 }
```

Listing 1 – Software Malha Aberta 1 Área

APÊNDICE B

```

1 // DEV_TCC_UFSC_ENC
2 // LEONARDO T FRANCIS
3 // Teste em Malha Aberta 2 Areas
4 #include<Math.h>          //Para utilizacao do Round
5 #include<SoftwareSerial.h> //Para utilizar o Meguno
6 #include<MegunoLink.h>    //Para utilizar o Meguno
7
8 TimePlot   MyPlot;
9
10 /* Portas PWM */
11 const int pwmPin2 = 2;
12 const int pwmPin3 = 3;
13 const int pwmPin4 = 4;
14 const int pwmPin5 = 5;
15
16 /* Portas Analogicas Sensores */
17 const int li_sensorPin0 = 0;
18 const int li_sensorPin1 = 1;
19 const int li_sensorPin2 = 2;
20 const int li_sensorPin3 = 3;
21
22 /* Variaveis Sensores e Lux */
23 double ld_sensor0_old, ld_sensor1_old, ld_sensor2_old, ld_sensor3_old;
24 double ld_lux_area01_sf, ld_lux_area23_sf;
25
26 // Variaveis Tempo
27 double ll_s0 = 0, ll_s1 = 0, ll_s2 = 0, ll_s3 = 0;
28 double tempo_inicio0=0, tempo_inicio1=0, tempo_inicio2=0, tempo_inicio3=0;
29 double tempo_fim0   =0, tempo_fim1   =0, tempo_fim2   =0, tempo_fim3   =0;
30
31 void setup() {
32   Serial.begin(9600); //Saida Output Serial
33 }
34
35 void loop() {
36   f_start();          // Start
37 }
38
39 void f_start()

```

```
40 {
41
42     analogWrite (pwmPin2, 125);
43     analogWrite (pwmPin3, 125);
44     analogWrite (pwmPin4, 125);
45     analogWrite (pwmPin5, 125);
46
47     if (ll_s0 == 0)
48     {
49         tempo_inicio0 = micros();
50         ld_sensor0_old = analogRead(li_sensorPin0);
51         ll_s0 = 1;
52     }
53     else
54     {
55         tempo_fim0 = micros();
56         ll_s0 = 0 ;
57     }
58
59     if (ll_s1 == 0)
60     {
61         tempo_inicio1 = micros();
62         ld_sensor1_old = analogRead(li_sensorPin1);
63         ll_s1 = 1;
64     }
65     else
66     {
67         tempo_fim1 = micros();
68         ll_s1 = 0 ;
69     }
70
71     if (ll_s2 == 0)
72     {
73         tempo_inicio2 = micros();
74         ld_sensor2_old = analogRead(li_sensorPin2);
75         ll_s2 = 1;
76     }
77     else
78     {
79         tempo_fim2 = micros();
80         ll_s2 = 0 ;
81     }
82
83     if (ll_s3 == 0)
84     {
85         tempo_inicio3 = micros();
86         ld_sensor3_old = analogRead(li_sensorPin3);
```

```
87     ll_s3 = 1;
88 }
89 else
90 {
91     tempo_fim3 = micros();
92     ll_s3 = 0;
93 }
94
95 if ( ll_s0 == 0 )
96 {
97     Serial.print("A0_Time: ");
98     Serial.print(tempo_fim0    tempo_inicio0);
99 }
100 if ( ll_s1 == 0 )
101 {
102     Serial.print("A1_Time: ");
103     Serial.print(tempo_fim1    tempo_inicio1);
104 }
105 if ( ll_s2 == 0 )
106 {
107     Serial.print("A2_Time: ");
108     Serial.print(tempo_fim2    tempo_inicio2);
109 }
110 if ( ll_s3 == 0 )
111 {
112     Serial.print("A3_Time: ");
113     Serial.print(tempo_fim3    tempo_inicio3);
114 }
115
116 // Media AD Temporario
117 Id_lux_area01_sf = (Id_sensor0_old + Id_sensor1_old )/2;
118 Id_lux_area23_sf = (Id_sensor2_old + Id_sensor3_old )/2;
119
120 MyPlot.SendData("AD_01_sf", Id_lux_area01_sf);
121 MyPlot.SendData("AD_23_sf", Id_lux_area23_sf);
122
123 }
```

Listing 2 – Software Malha Aberta 2 Área

APÊNDICE C

```

1 // DEV_TCC_UFSC_ENC
2 // LEONARDO T FRANCIS
3 // Teste em Malha Aberta 4 Areas
4 #include <Math.h> //Para realizar calculos
5 #include<SoftwareSerial.h> //Para utilizar o Meguno
6 #include<MegunoLink.h> //Para utilizar o Meguno
7
8 // Portas PWM
9 const int pwmPin2 = 2;
10 const int pwmPin3 = 3;
11 const int pwmPin4 = 4;
12 const int pwmPin5 = 5;
13 const int pwmPin8 = 8;
14 const int pwmPin9 = 9;
15
16 // Portas Analogicas Sensores
17 const int li_sensorPin0 = 0;
18 const int li_sensorPin1 = 1;
19 const int li_sensorPin2 = 2;
20 const int li_sensorPin3 = 3;
21
22 // Variaveis Sensores e Lux
23 double ld_sensor0_old , ld_sensor1_old , ld_sensor2_old , ld_sensor3_old ;
24
25 // Variaveis Tempo
26 double ll_s0 = 0, ll_s1 = 0, ll_s2 = 0, ll_s3 = 0;
27 double tempo_inicio0=0, tempo_inicio1=0, tempo_inicio2=0, tempo_inicio3=0;
28 double tempo_fim0 =0, tempo_fim1 =0, tempo_fim2 =0, tempo_fim3 =0;
29
30 TimePlot MyPlot;
31
32 void setup() {
33   Serial.begin(9600);
34 }
35
36 void loop() {
37   f_start(); // Start
38 }
39

```

```
40 void f_start()
41 {
42     analogWrite(pwmPin2, 125);
43     analogWrite(pwmPin3, 125);
44     analogWrite(pwmPin4, 125);
45     analogWrite(pwmPin5, 125);
46
47     if (ll_s0 == 0)
48     {
49         tempo_inicio0 = micros();
50         ld_sensor0_old = analogRead(li_sensorPin0);
51         ll_s0 = 1;
52     }
53     else
54     {
55         tempo_fim0 = micros();
56         ll_s0 = 0 ;
57     }
58
59     if (ll_s1 == 0)
60     {
61         tempo_inicio1 = micros();
62         ld_sensor1_old = analogRead(li_sensorPin1);
63         ll_s1 = 1;
64     }
65     else
66     {
67         tempo_fim1 = micros();
68         ll_s1 = 0 ;
69     }
70
71     if (ll_s2 == 0)
72     {
73         tempo_inicio2 = micros();
74         ld_sensor2_old = analogRead(li_sensorPin2);
75         ll_s2 = 1;
76     }
77     else
78     {
79         tempo_fim2 = micros();
80         ll_s2 = 0 ;
81     }
82
83     if (ll_s3 == 0)
84     {
85         tempo_inicio3 = micros();
86         ld_sensor3_old = analogRead(li_sensorPin3);
```

```
87     ll_s3 = 1;
88 }
89 else
90 {
91     tempo_fim3 = micros();
92     ll_s3 = 0;
93 }
94
95 if ( ll_s0 == 0 )
96 {
97     Serial.print("A0_Time: ");
98     Serial.print(tempo_fim0    tempo_inicio0);
99 }
100 if ( ll_s1 == 0 )
101 {
102     Serial.print("A1_Time: ");
103     Serial.print(tempo_fim1    tempo_inicio1);
104 }
105 if ( ll_s2 == 0 )
106 {
107     Serial.print("A2_Time: ");
108     Serial.print(tempo_fim2    tempo_inicio2);
109 }
110 if ( ll_s3 == 0 )
111 {
112     Serial.print("A3_Time: ");
113     Serial.print(tempo_fim3    tempo_inicio3);
114 }
115
116 /* Output */
117 MyPlot.SendData("AD_0_cf", Id_sensor0_old);
118 MyPlot.SendData("AD_1_cf", Id_sensor1_old);
119 MyPlot.SendData("AD_2_cf", Id_sensor2_old);
120 MyPlot.SendData("AD_3_cf", Id_sensor3_old);
121 }
```

Listing 3 – Software Malha Aberta 4 Área

APÊNDICE D

```
1 // DEV_TCC_UFSC_ENC
2 // LEONARDO T FRANCIS
3 // Teste em Malha fechada, com todas as funcoes realizadas
4 #include <Math.h> //Para utilizacao do Round
5 #include <PID_v1.h> //Inclusao da biblioteca PID.
6 #include <RTCLib.h> //Para utilizar RTC
7
8 #include<SoftwareSerial.h> //Para utilizar o Meguno
9 #include<MegunoLink.h> //Para utilizar o Meguno
10
11
12 //Funcoes finalizadas 11/05/2019
13 double a0= 7037.3357143071834798320196568966;
14 double a1 = 23.176922517619740204963818541728;
15 double a2 = 0.026336783720252783524973239082101;
16 double a3 = 0.000010179015249403274099432303856361;
17 double a4 = 0;
18 double a5 = 0;
19 double a6 = 0;
20 double a7 = 0;
21 double a8 = 0;
22
23 double b0 = 9611.9077345140431134495884180069;
24 double b1 = 27.560368784496009908480118610896;
25 double b2 = 0.027194777160568516655159143624587;
26 double b3 = 0.0000092283476468056846670276471833461;
27 double b4 = 0;
28 double b5 = 0;
29 double b6 = 0;
30 double b7 = 0;
31 double b8 = 0;
32
33 double c0 = 5823.0588701080050668679177761078;
34 double c1 = 18.641985550191158438337879488245;
35 double c2 = 0.020321971906241263761128124087918;
36 double c3 = 0.0000074990756668287077553415251318647;
37 double c4 = 0;
38 double c5 = 0;
39 double c6 = 0;
```

```
40 double c7 = 0;
41 double c8 = 0;
42
43 double d0 = 13283.851758887467440217733383179;
44 double d1 = 78.071211489245072812082071322948;
45 double d2 = 0.22030259097055332473402700088627;
46 double d3 = 0.00038640813948037243310412724994762;
47 double d4 = 0.00000044798470122337224795931493077916;
48 double d5 = 0.00000000033484517457280024532090364458689;
49 double d6 = 0.00000000000014542219758720604784972818753156;
50 double d7 = 0.000000000000000027666723628194782196854880863473;
51 double d8 = 0;
52
53 /* Portas PWM */
54 const int pwmPin2 = 2;
55 const int pwmPin3 = 3;
56 const int pwmPin4 = 4;
57 const int pwmPin5 = 5;
58
59 /* Portas Digitais LEDs */
60 const int ledD4 = 27;
61 const int ledD3 = 29;
62 const int ledD2 = 31;
63 const int ledD1 = 33;
64
65 /* Portas Digitais Botoes */
66 const int Button0 = 26;
67 const int Button1 = 28;
68 const int Button2 = 30;
69 const int Button3 = 32;
70
71 /* Portas Analogicas Sensores */
72 const int li_sensorPin0 = 0;
73 const int li_sensorPin1 = 1;
74 const int li_sensorPin2 = 2;
75 const int li_sensorPin3 = 3;
76
77 /* Variaveis Sensores e Lux */
78 double Id_sensor0 , Id_sensor1 , Id_sensor2 , Id_sensor3;
79 double Id_sensor0_old , Id_sensor1_old , Id_sensor2_old , Id_sensor3_old;
80 double Id_sensor0_ef , Id_sensor1_ef , Id_sensor2_ef , Id_sensor3_ef;
81
82 double Id_sensor0123_old , Id_sensor0123;
83
84 double Id_lux_area0123_sf , Id_lux_area0123_cf , Id_lux_area0123_ef;
85
86 // Control Values
```

```

87 double referencia , entrada_0123 , saida_0123; //Declaracao das variaveis
    referencia , entrada e saida.
88 double Kp=0.075, Ki=0.30, Kd=0; //Declaracao e definicao das
    constantes do controlador PID
89
90 //Exponential Filter
91 double S_t_1 = 0, S_t_0 = 0,S_t_2 = 0,S_t_3 = 0, X_t_0 = 0,X_t_1 = 0,X_t_2
    = 0,X_t_3 = 0,S_t1_0 = 0, S_t1_1 = 0, S_t1_2 = 0,S_t1_3 = 0;
92 const double alp = 0.05;
93
94 /* RTC */
95 RTC_DS3231 rtc;
96
97 double Il_s0 = 0, Il_s1 = 0, Il_s2 = 0, Il_s3 = 0;
98 double tempo_inicio0=0, tempo_inicio1=0, tempo_inicio2=0, tempo_inicio3=0;
99 double tempo_fim0 =0, tempo_fim1 =0, tempo_fim2 =0, tempo_fim3 =0;
100
101 PID PID_Area0123(&entrada_0123 , &saida_0123 , &referencia , Kp, Ki, Kd,
    DIRECT); //Chamada da classe PID sendo "myPID" uma funcao do tipo PID.
    //REVERSE = PARA AD, DIRECT = LUX
102
103 TimePlot MyPlot;
104
105 void setup() {
106     Serial.begin(9600); //Saida OutputSerial
107
108     referencia = 480; //Passa o valor desejado como referencia. //
        REFERENCIA = 600 [AD] REFERENCIA = 480 [LUX]
109
110     PID_Area0123.SetMode(AUTOMATIC);
111
112     pinMode(ledD1, OUTPUT);
113     pinMode(ledD2, OUTPUT);
114     pinMode(ledD3, OUTPUT);
115     pinMode(ledD4, OUTPUT);
116 }
117
118 void loop() {
119
120     DateTime ldt_now = rtc.now();
121
122     if (! rtc.begin()) {
123         Serial.println("Problema com RTC");
124         while (1);
125     }
126
127     if (rtc.lostPower()) {

```

```
128     Serial.println("RTC lost power, setting the time!");
129     rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
130 }
131
132 if ((long(ldt_now.hour()) <= 23 and long(ldt_now.minute()) <= 59) and (
133     long(ldt_now.hour()) >= 6 and long(ldt_now.minute()) >= 0))
134 {
135     f_status_system(1); // Inicializa
136     f_start();         // Start
137 }
138 else
139 {
140     f_status_system(0); // Stop
141 }
142
143
144 void f_status_system(int ls_status)
145 {
146     if (ls_status == 1){
147         referencia = 480; // Start Control System
148         digitalWrite(ledD1, LOW);
149     }
150     else{
151         referencia = 0; // Stop Control System
152         digitalWrite(ledD1, HIGH);
153     }
154
155     digitalWrite(ledD2, HIGH);
156     digitalWrite(ledD3, HIGH);
157     digitalWrite(ledD4, HIGH);
158 }
159
160 void f_start()
161 {
162
163     ld_sensor0_old = analogRead(li_sensorPin0);
164     ld_sensor1_old = analogRead(li_sensorPin1);
165     ld_sensor2_old = analogRead(li_sensorPin2);
166     ld_sensor3_old = analogRead(li_sensorPin3);
167
168     // fc = 0.10
169     ld_sensor0 = 0.9327*ld_sensor0+0.06732*ld_sensor0_old;
170     ld_sensor1 = 0.9327*ld_sensor1+0.06732*ld_sensor1_old;
171     ld_sensor2 = 0.9327*ld_sensor2+0.06732*ld_sensor2_old;
172     ld_sensor3 = 0.9327*ld_sensor3+0.06732*ld_sensor3_old;
173
```

```

174 //Aplicacao do filtro Exponencial nos valores em AD de entrada
175 ld_sensor0_ef = f_exponential_filter(ld_sensor0_old, '0');
176 ld_sensor1_ef = f_exponential_filter(ld_sensor1_old, '1');
177 ld_sensor2_ef = f_exponential_filter(ld_sensor2_old, '2');
178 ld_sensor3_ef = f_exponential_filter(ld_sensor3_old, '3');
179
180 //Conversao AD LUX + Media
181 ld_lux_area0123_cf = (f_ad_to_lux(ld_sensor0, a0, a1, a2, a3, a4, a5,
    a6, a7, a8, '0') + f_ad_to_lux(ld_sensor1, b0, b1, b2, b3, b4, b5,
    b6, b7, b8, '1') + f_ad_to_lux(ld_sensor2, c0, c1, c2, c3, c4, c5,
    c6, c7, c8, '2') + f_ad_to_lux(ld_sensor3, d0, d1, d2, d3, d4, d5,
    d6, d7, d8, '3'))/4;
182 ld_lux_area0123_sf = (f_ad_to_lux(ld_sensor0_old, a0, a1, a2, a3, a4,
    a5, a6, a7, a8, '0') + f_ad_to_lux(ld_sensor1_old, b0, b1, b2, b3,
    b4, b5, b6, b7, b8, '1') + f_ad_to_lux(ld_sensor2_old, c0, c1, c2,
    c3, c4, c5, c6, c7, c8, '2') + f_ad_to_lux(ld_sensor3_old, d0, d1,
    d2, d3, d4, d5, d6, d7, d8, '3'))/4;
183
184 ld_lux_area0123_ef = (f_ad_to_lux(ld_sensor0_ef, a0, a1, a2, a3, a4, a5
    , a6, a7, a8, '0') + f_ad_to_lux(ld_sensor1_ef, b0, b1, b2, b3, b4,
    b5, b6, b7, b8, '1') + f_ad_to_lux(ld_sensor2_ef, c0, c1, c2, c3, c4
    , c5, c6, c7, c8, '2') + f_ad_to_lux(ld_sensor3_ef, d0, d1, d2, d3,
    d4, d5, d6, d7, d8, '3'))/4;
185
186 // Valor de entrada sera o AD Filtrado
187 entrada_0123 = ld_lux_area0123_cf;
188
189 PID_Area0123.Compute();
190
191 analogWrite(pwmPin2, saida_0123);
192 analogWrite(pwmPin3, saida_0123);
193 analogWrite(pwmPin4, saida_0123);
194 analogWrite(pwmPin5, saida_0123);
195
196 MyPlot.SendData("AD_sf", ld_lux_area0123_sf);
197 MyPlot.SendData("AD_cf", ld_lux_area0123_cf);
198 MyPlot.SendData("AD_expo_f", ld_lux_area0123_ef);
199
200 MyPlot.SendData("SAIDA_AREA0123_CF(%)", f_percent_saida(saida_0123));
201
202 delay(1000);
203 }
204
205
206 long f_ad_to_lux(double ld_ad_sensor, double a0, double a1, double a2,
    double a3, double a4, double a5, double a6, double a7, double a8, char
    ls_tipo)

```

```
207 {
208     long ll_return = 0;
209
210     if ((ls_tipo == '0' and ld_ad_sensor > 1006) or (ls_tipo == '1' and
        ld_ad_sensor > 1020) or (ls_tipo == '2' and ld_ad_sensor > 1003) or
        (ls_tipo == '3' and ld_ad_sensor > 1022))
211     {
212         return 0;
213     }
214
215     ll_return = a8*pow(ld_ad_sensor,8) + a7*pow(ld_ad_sensor,7) + a6*pow(
        ld_ad_sensor,6) + a5*pow(ld_ad_sensor,5) + a4*pow(ld_ad_sensor,4) +
        a3*pow(ld_ad_sensor,3) + a2*pow(ld_ad_sensor,2)+ a1*ld_ad_sensor +
        a0;
216
217     return ll_return;
218 }
219
220
221 double f_exponential_filter(double ld_ad_sensor, char ls_tipo)
222 {
223     double ll_return = 0;
224
225     if (ls_tipo == '0')
226     {
227         X_t_0 = ld_ad_sensor * 5.0/1023.0;
228
229         S_t_0 = alp * X_t_0 + (1 - alp) * S_t1_0;
230
231         S_t1_0 = S_t_0;
232
233         ll_return = S_t_0;
234     }
235     else if (ls_tipo == '1')
236     {
237         X_t_1 = ld_ad_sensor * 5.0/1023.0;
238
239         S_t_1 = alp * X_t_1 + (1 - alp) * S_t1_1;
240
241         S_t1_1 = S_t_1;
242
243         ll_return = S_t_1;
244     }
245     else if (ls_tipo == '2')
246     {
247         X_t_2 = ld_ad_sensor * 5.0/1023.0;
248
```

```
249     S_t_2 = alp * X_t_2 + (1 - alp) * S_t1_2;
250
251     S_t1_2 = S_t_2;
252
253     ll_return = S_t_2;
254 }
255 else
256 {
257     X_t_3 = Id_ad_sensor * 5.0/1023.0;
258
259     S_t_3 = alp * X_t_3 + (1 - alp) * S_t1_3;
260
261     S_t1_3 = S_t_3;
262
263     ll_return = S_t_3;
264 }
265
266
267     ll_return = ll_return * 1023.0/5.0;
268
269     return ll_return;
270 }
271
272 long f_percent_saida(double Id_saida)
273 {
274     long ll_pr_uso;
275
276     ll_pr_uso = (Id_saida*100)/255;
277
278     return ll_pr_uso;
279 }
```

Listing 4 – Software Malha Fechada 1 Área

APÊNDICE E

```
1 // DEV_TCC_UFSC_ENC
2 // LEONARDO T FRANCIS
3 // Teste em Malha Fechada, com todas as funcoes realizadas
4 #include <Math.h> //Para utilizacao do Round
5 #include <PID_v1.h> //Inclusao da biblioteca PID.
6 #include <RTCLib.h> //Para utilizar RTC
7
8 #include<SoftwareSerial.h> //Para utilizar o Meguno
9 #include<MegunoLink.h> //Para utilizar o Meguno
10
11 //Funcoes finalizadas 11/05/2019
12 double a0= 7037.3357143071834798320196568966;
13 double a1 = 23.176922517619740204963818541728;
14 double a2 = 0.026336783720252783524973239082101;
15 double a3 = 0.000010179015249403274099432303856361;
16 double a4 = 0;
17 double a5 = 0;
18 double a6 = 0;
19 double a7 = 0;
20 double a8 = 0;
21
22 double b0 = 9611.9077345140431134495884180069;
23 double b1 = 27.560368784496009908480118610896;
24 double b2 = 0.027194777160568516655159143624587;
25 double b3 = 0.0000092283476468056846670276471833461;
26 double b4 = 0;
27 double b5 = 0;
28 double b6 = 0;
29 double b7 = 0;
30 double b8 = 0;
31
32 double c0 = 5823.0588701080050668679177761078;
33 double c1 = 18.641985550191158438337879488245;
34 double c2 = 0.020321971906241263761128124087918;
35 double c3 = 0.0000074990756668287077553415251318647;
36 double c4 = 0;
37 double c5 = 0;
38 double c6 = 0;
39 double c7 = 0;
```

```
40 double c8 = 0;
41
42 double d0 = 13283.851758887467440217733383179;
43 double d1 = 78.071211489245072812082071322948;
44 double d2 = 0.22030259097055332473402700088627;
45 double d3 = 0.00038640813948037243310412724994762;
46 double d4 = 0.00000044798470122337224795931493077916;
47 double d5 = 0.00000000033484517457280024532090364458689;
48 double d6 = 0.00000000000014542219758720604784972818753156;
49 double d7 = 0.000000000000000027666723628194782196854880863473;
50 double d8 = 0;
51
52 /* Portas PWM */
53 const int pwmPin2 = 2;
54 const int pwmPin3 = 3;
55 const int pwmPin4 = 4;
56 const int pwmPin5 = 5;
57
58 /* Portas Digitais LEDs */
59 const int ledD4 = 27;
60 const int ledD3 = 29;
61 const int ledD2 = 31;
62 const int ledD1 = 33;
63
64 /* Portas Digitais Botoes */
65 const int Button0 = 26;
66 const int Button1 = 28;
67 const int Button2 = 30;
68 const int Button3 = 32;
69
70 /* Portas Analogicas Sensores */
71 const int li_sensorPin0 = 0;
72 const int li_sensorPin1 = 1;
73 const int li_sensorPin2 = 2;
74 const int li_sensorPin3 = 3;
75
76 /* Variaveis Sensores e Lux */
77 double ld_sensor0, ld_sensor1, ld_sensor2, ld_sensor3;
78 double ld_sensor0_ef, ld_sensor1_ef, ld_sensor2_ef, ld_sensor3_ef;
79 double ld_sensor0_old, ld_sensor1_old, ld_sensor2_old, ld_sensor3_old;
80
81 double ld_sensor01_old, ld_sensor23_old, ld_sensor01, ld_sensor23;
82
83 double ld_lux_area01_sf, ld_lux_area23_sf;
84 double ld_lux_area01_cf, ld_lux_area23_cf;
85 double ld_lux_area01_ef, ld_lux_area23_ef;
86
```

```

87 // Variaveis Tempo
88 double I1_s0 = 0, I1_s1 = 0, I1_s2 = 0, I1_s3 = 0;
89 double tempo_inicio0=0, tempo_inicio1=0, tempo_inicio2=0, tempo_inicio3=0;
90 double tempo_fim0    =0, tempo_fim1    =0, tempo_fim2    =0, tempo_fim3    =0;
91
92 // Control Values
93 double referencia , entrada_01, entrada_23, saida_01, saida_23; //
    Declaracao das variaveis referencia , entrada e saida.
94 double Kp=0.075, Ki=0.30, Kd=0;          //Kp=0.075, Ki=0.30, Kd=0
                                           //Declaracao
    e definicao das constantes do controlador PID
95
96 // Exponential Filter
97 double S_t_1 = 0, S_t_0 = 0, S_t_2 = 0, S_t_3 = 0, X_t_0 = 0, X_t_1 = 0, X_t_2
    = 0, X_t_3 = 0, S_t1_0 = 0, S_t1_1 = 0, S_t1_2 = 0, S_t1_3 = 0;
98 const double alp = 0.05;
99
100 RTC_DS3231 rtc;
101
102 PID PID_Area01(&entrada_01, &saida_01, &referencia, Kp, Ki, Kd, DIRECT); //
    Chamada da classe PID sendo "myPID" uma funcao do tipo PID.
103 PID PID_Area23(&entrada_23, &saida_23, &referencia, Kp, Ki, Kd, DIRECT); //
    Chamada da classe PID sendo "myPID" uma funcao do tipo PID.
104
105 TimePlot MyPlot;
106
107 void setup() {
108     Serial.begin(9600); // Saida Output Serial
109
110     referencia = 480; // Passa o valor desejado como referencia. //
        REFERENCIA = 600 [AD] REFERENCIA = 480 [LUX]
111
112     PID_Area01.SetMode(AUTOMATIC);
113     PID_Area23.SetMode(AUTOMATIC);
114
115     pinMode(ledD1, OUTPUT);
116     pinMode(ledD2, OUTPUT);
117     pinMode(ledD3, OUTPUT);
118     pinMode(ledD4, OUTPUT);
119 }
120
121 void loop() {
122     DateTime ldt_now = rtc.now();
123
124     if (! rtc.begin()) {
125         Serial.println("Problema com RTC");
126         while (1);

```

```
127     }
128
129     if (rtc.lostPower()) {
130         Serial.println("RTC lost power, setting the time!");
131         rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
132     }
133
134     if ((long(ldt_now.hour()) <= 23 and long(ldt_now.minute()) <= 59) and (
135         long(ldt_now.hour()) >= 6 and long(ldt_now.minute()) >= 0))
136     {
137         f_status_system(1); // Inicializa
138         f_start();          // Start
139     }
140     else
141     {
142         f_status_system(0); // Stop
143     }
144 }
145 void f_status_system(int ls_status)
146 {
147     if (ls_status == 1){
148         referencia = 480;      // Start Control System
149         digitalWrite(ledD1, LOW);
150     }
151     else{
152         referencia = 0;       // Stop Control System
153         digitalWrite(ledD1, HIGH);
154     }
155
156     digitalWrite(ledD2, HIGH);
157     digitalWrite(ledD3, HIGH);
158     digitalWrite(ledD4, HIGH);
159 }
160
161 void f_start()
162 {
163
164     Id_sensor0_old = analogRead(li_sensorPin0);
165     Id_sensor1_old = analogRead(li_sensorPin1);
166     Id_sensor2_old = analogRead(li_sensorPin2);
167     Id_sensor3_old = analogRead(li_sensorPin3);
168
169     //Aplica Filtro 1st Order cada Sensor
170
171     // fc = 0.05
172     Id_sensor0 = 0.9319*Id_sensor0+0.06814*Id_sensor0_old;
```

```

173 ld_sensor1 = 0.9319*ld_sensor1+0.06814*ld_sensor1_old;
174 ld_sensor2 = 0.9319*ld_sensor2+0.06814*ld_sensor2_old;
175 ld_sensor3 = 0.9319*ld_sensor3+0.06814*ld_sensor3_old;
176
177 //Aplicacao do filtro Exponencial nos valores em AD de entrada
178 ld_sensor0_ef = f_exponential_filter(ld_sensor0_old, '0');
179 ld_sensor1_ef = f_exponential_filter(ld_sensor1_old, '1');
180 ld_sensor2_ef = f_exponential_filter(ld_sensor2_old, '2');
181 ld_sensor3_ef = f_exponential_filter(ld_sensor3_old, '3');
182
183 //Conversao AD LUX + Media area
184 ld_lux_area01_cf = (f_ad_to_lux(ld_sensor0, a0, a1, a2, a3, a4, a5, a6,
    a7, a8, '0') + f_ad_to_lux(ld_sensor1, b0, b1, b2, b3, b4, b5, b6,
    b7, b8, '1'))/2;
185 ld_lux_area23_cf = (f_ad_to_lux(ld_sensor2, c0, c1, c2, c3, c4, c5, c6,
    c7, c8, '2') + f_ad_to_lux(ld_sensor3, d0, d1, d2, d3, d4, d5, d6,
    d7, d8, '3'))/2;
186
187 ld_lux_area01_sf = (f_ad_to_lux(ld_sensor0_old, a0, a1, a2, a3, a4, a5,
    a6, a7, a8, '0') + f_ad_to_lux(ld_sensor1_old, b0, b1, b2, b3, b4,
    b5, b6, b7, b8, '1'))/2;
188 ld_lux_area23_sf = (f_ad_to_lux(ld_sensor2_old, c0, c1, c2, c3, c4, c5,
    c6, c7, c8, '2') + f_ad_to_lux(ld_sensor3_old, d0, d1, d2, d3, d4,
    d5, d6, d7, d8, '3'))/2;
189
190 ld_lux_area01_ef = (f_ad_to_lux(ld_sensor0_ef, a0, a1, a2, a3, a4, a5,
    a6, a7, a8, '0') + f_ad_to_lux(ld_sensor1_ef, b0, b1, b2, b3, b4, b5
    , b6, b7, b8, '1'))/2;
191 ld_lux_area23_ef = (f_ad_to_lux(ld_sensor2_ef, c0, c1, c2, c3, c4, c5,
    c6, c7, c8, '2') + f_ad_to_lux(ld_sensor3_ef, d0, d1, d2, d3, d4, d5
    , d6, d7, d8, '3'))/2;
192
193 // Valor de entrada sera o Lux Filtrado
194 entrada_01 = ld_lux_area01_cf;
195 entrada_23 = ld_lux_area23_cf;
196
197 PID_Area01.Compute();
198 PID_Area23.Compute();
199
200 /* Saida PWM */
201 analogWrite(pwmPin2, saida_01);
202 analogWrite(pwmPin3, saida_01);
203 analogWrite(pwmPin4, saida_23);
204 analogWrite(pwmPin5, saida_23);
205
206 MyPlot.SendData("AD_01_sf", ld_lux_area01_sf);
207 MyPlot.SendData("AD_23_sf", ld_lux_area23_sf);

```

```

208 MyPlot.SendData("AD_01_cf", Id_lux_area01_cf);
209 MyPlot.SendData("AD_23_cf", Id_lux_area23_cf);
210 MyPlot.SendData("AD_01_ef", Id_lux_area01_ef);
211 MyPlot.SendData("AD_23_ef", Id_lux_area23_ef);
212
213 MyPlot.SendData("SAIDA_AREA01(%)", f_percent_saida(saida_01));
214 MyPlot.SendData("SAIDA_AREA23(%)", f_percent_saida(saida_23));
215
216 delay(1000);
217 }
218
219
220 long f_ad_to_lux(double Id_ad_sensor, double a0, double a1, double a2,
double a3, double a4, double a5, double a6, double a7, double a8, char
ls_tipo)
221 {
222     long ll_return = 0;
223
224     if ((ls_tipo == '0' and Id_ad_sensor > 1006) or (ls_tipo == '1' and
        Id_ad_sensor > 1020) or (ls_tipo == '2' and Id_ad_sensor > 1003) or
        (ls_tipo == '3' and Id_ad_sensor > 1022))
225     {
226         return 0;
227     }
228
229     ll_return = a8*pow(Id_ad_sensor,8) + a7*pow(Id_ad_sensor,7) + a6*pow(
        Id_ad_sensor,6) + a5*pow(Id_ad_sensor,5) + a4*pow(Id_ad_sensor,4) +
        a3*pow(Id_ad_sensor,3) + a2*pow(Id_ad_sensor,2)+ a1*Id_ad_sensor +
        a0;
230
231     return ll_return;
232 }
233
234 double f_exponential_filter(double Id_ad_sensor, char ls_tipo)
235 {
236     double ll_return = 0;
237
238     if (ls_tipo == '0')
239     {
240         X_t_0 = Id_ad_sensor * 5.0/1023.0;
241
242         S_t_0 = alp * X_t_0 + (1 - alp) * S_t1_0;
243
244         S_t1_0 = S_t_0;
245
246         ll_return = S_t_0;
247     }

```

```
248     else if (ls_tipo == '1')
249     {
250         X_t_1 = ld_ad_sensor * 5.0/1023.0;
251
252         S_t_1 = alp * X_t_1 + (1 - alp) * S_t1_1;
253
254         S_t1_1 = S_t_1;
255
256         ll_return = S_t_1;
257     }
258     else if (ls_tipo == '2')
259     {
260         X_t_2 = ld_ad_sensor * 5.0/1023.0;
261
262         S_t_2 = alp * X_t_2 + (1 - alp) * S_t1_2;
263
264         S_t1_2 = S_t_2;
265
266         ll_return = S_t_2;
267     }
268     else
269     {
270         X_t_3 = ld_ad_sensor * 5.0/1023.0;
271
272         S_t_3 = alp * X_t_3 + (1 - alp) * S_t1_3;
273
274         S_t1_3 = S_t_3;
275
276         ll_return = S_t_3;
277     }
278
279
280     ll_return = ll_return *1023.0/5.0;
281
282     return ll_return;
283 }
284
285 long f_percent_saida(double ld_saida)
286 {
287     long ll_pr_uso;
288
289     ll_pr_uso = (ld_saida*100)/255;
290
291     return ll_pr_uso;
292 }
```

Listing 5 – Software Malha Fechada 2 Áreas

APÊNDICE F

```
1
2 // DEV_TCC_UFSC_ENC
3 // LEONARDO T FRANCIS
4 // Teste em Malha fechada, com todas as funcoes realizadas
5 #include <Math.h> //Para utilizacao do Round
6 #include <PID_v1.h> //Inclusao da biblioteca PID.
7 //#include <RTCLib.h> //Para utilizar RTC
8 //#include <DS3231.h> //Para utilizar RTC
9
10 #include<SoftwareSerial.h> //Para utilizar o Meguno
11 #include<MegunoLink.h> //Para utilizar o Meguno
12
13 //Funcoes finalizadas 11/05/2019
14 double a0= 7037.3357143071834798320196568966;
15 double a1 = 23.176922517619740204963818541728;
16 double a2 = 0.026336783720252783524973239082101;
17 double a3 = 0.000010179015249403274099432303856361;
18 double a4 = 0;
19 double a5 = 0;
20 double a6 = 0;
21 double a7 = 0;
22 double a8 = 0;
23
24 double b0 = 9611.9077345140431134495884180069;
25 double b1 = 27.560368784496009908480118610896;
26 double b2 = 0.027194777160568516655159143624587;
27 double b3 = 0.0000092283476468056846670276471833461;
28 double b4 = 0;
29 double b5 = 0;
30 double b6 = 0;
31 double b7 = 0;
32 double b8 = 0;
33
34 double c0 = 5823.0588701080050668679177761078;
35 double c1 = 18.641985550191158438337879488245;
36 double c2 = 0.020321971906241263761128124087918;
37 double c3 = 0.0000074990756668287077553415251318647;
38 double c4 = 0;
39 double c5 = 0;
```

```

40 double c6 = 0;
41 double c7 = 0;
42 double c8 = 0;
43
44 double d0 = 13283.851758887467440217733383179;
45 double d1 = 78.071211489245072812082071322948;
46 double d2 = 0.22030259097055332473402700088627;
47 double d3 = 0.00038640813948037243310412724994762;
48 double d4 = 0.00000044798470122337224795931493077916;
49 double d5 = 0.00000000033484517457280024532090364458689;
50 double d6 = 0.0000000000014542219758720604784972818753156;
51 double d7 = 0.0000000000000000027666723628194782196854880863473;
52 double d8 = 0;
53
54 /* Portas PWM */
55 const int pwmPin2 = 2;
56 const int pwmPin3 = 3;
57 const int pwmPin4 = 4;
58 const int pwmPin5 = 5;
59
60 /* Portas Analogicas Sensores */
61 const int li_sensorPin0 = 0;
62 const int li_sensorPin1 = 1;
63 const int li_sensorPin2 = 2;
64 const int li_sensorPin3 = 3;
65
66 /* Variaveis Sensores e Lux */
67 double ld_sensor0, ld_sensor1, ld_sensor2, ld_sensor3;
68 double ld_sensor0_ef, ld_sensor1_ef, ld_sensor2_ef, ld_sensor3_ef;
69 double ld_sensor0_old, ld_sensor1_old, ld_sensor2_old, ld_sensor3_old;
70
71 double ld_lux_area0_sf, ld_lux_area1_sf, ld_lux_area2_sf, ld_lux_area3_sf;
72 double ld_lux_area0_cf, ld_lux_area1_cf, ld_lux_area2_cf, ld_lux_area3_cf;
73 double ld_lux_area0_ef, ld_lux_area1_ef, ld_lux_area2_ef, ld_lux_area3_ef;
74
75 // Variaveis Tempo
76 double ll_s0 = 0, ll_s1 = 0, ll_s2 = 0, ll_s3 = 0;
77 double tempo_inicio0=0, tempo_inicio1=0, tempo_inicio2=0, tempo_inicio3=0;
78 double tempo_fim0 =0, tempo_fim1 =0, tempo_fim2 =0, tempo_fim3 =0;
79
80 // Control Values
81 double referencia, entrada_0, entrada_1, entrada_2, entrada_3, saida_0,
    saida_1, saida_2, saida_3; //Declaracao das variaveis referencia,
    entrada e saida.
82 double Kp=0.075, Ki=0.30, Kd=0; //Kp=0.075, Ki=0.30, Kd=0
    //Declaracao
    e definicao das constantes do controlador PID

```

```

83
84 // Exponential Filter
85 double S_t_1 = 0, S_t_0 = 0, S_t_2 = 0, S_t_3 = 0, X_t_0 = 0, X_t_1 = 0, X_t_2
    = 0, X_t_3 = 0, S_t1_0 = 0, S_t1_1 = 0, S_t1_2 = 0, S_t1_3 = 0;
86 const double alp = 0.1;
87
88 PID PID_Area0(&entrada_0, &saida_0, &referencia, Kp, Ki, Kd, DIRECT); //
    Chamada da classe PID sendo "myPID" uma funcao do tipo PID.
89 PID PID_Area1(&entrada_1, &saida_1, &referencia, Kp, Ki, Kd, DIRECT); //
    Chamada da classe PID sendo "myPID" uma funcao do tipo PID.
90 PID PID_Area2(&entrada_2, &saida_2, &referencia, Kp, Ki, Kd, DIRECT); //
    Chamada da classe PID sendo "myPID" uma funcao do tipo PID.
91 PID PID_Area3(&entrada_3, &saida_3, &referencia, Kp, Ki, Kd, DIRECT); //
    Chamada da classe PID sendo "myPID" uma funcao do tipo PID.
92
93 TimePlot MyPlot;
94
95 void setup() {
96     Serial.begin(9600); // Saida Output Serial
97
98     referencia = 480; // Passa o valor desejado como referencia. //
        REFERENCIA = 600 [AD] REFERENCIA = 480 [LUX]
99
100     PID_Area0.SetMode(AUTOMATIC);
101     PID_Area1.SetMode(AUTOMATIC);
102     PID_Area2.SetMode(AUTOMATIC);
103     PID_Area3.SetMode(AUTOMATIC);
104 }
105
106 void loop() {
107     f_start(); // Start
108 }
109
110 void f_start()
111 {
112
113     Id_sensor0_old = analogRead(li_sensorPin0);
114     Id_sensor1_old = analogRead(li_sensorPin1);
115     Id_sensor2_old = analogRead(li_sensorPin2);
116     Id_sensor3_old = analogRead(li_sensorPin3);
117
118     // Aplica Filtro 1st Order cada Sensor
119     // T = 0.582400 fc = 0.05
120     Id_sensor0 = 0.8328*Id_sensor0+0.1672*Id_sensor0_old;
121     Id_sensor1 = 0.8328*Id_sensor1+0.1672*Id_sensor1_old;
122     Id_sensor2 = 0.8328*Id_sensor2+0.1672*Id_sensor2_old;
123     Id_sensor3 = 0.8328*Id_sensor3+0.1672*Id_sensor3_old;

```

```
124
125 //Aplicacao do filtro Exponencial nos valores em AD de entrada
126 ld_sensor0_ef = f_exponential_filter(ld_sensor0_old, '0');
127 ld_sensor1_ef = f_exponential_filter(ld_sensor1_old, '1');
128 ld_sensor2_ef = f_exponential_filter(ld_sensor2_old, '2');
129 ld_sensor3_ef = f_exponential_filter(ld_sensor3_old, '3');
130
131 //Conversao AD LUX + Media area
132 ld_lux_area0_cf = f_ad_to_lux(ld_sensor0, a0, a1, a2, a3, a4, a5, a6,
    a7, a8, '0');
133 ld_lux_area1_cf = f_ad_to_lux(ld_sensor1, b0, b1, b2, b3, b4, b5, b6,
    b7, b8, '1');
134 ld_lux_area2_cf = f_ad_to_lux(ld_sensor2, c0, c1, c2, c3, c4, c5, c6,
    c7, c8, '2');
135 ld_lux_area3_cf = f_ad_to_lux(ld_sensor3, d0, d1, d2, d3, d4, d5, d6,
    d7, d8, '3');
136
137 ld_lux_area0_sf = f_ad_to_lux(ld_sensor0_old, a0, a1, a2, a3, a4, a5,
    a6, a7, a8, '0');
138 ld_lux_area1_sf = f_ad_to_lux(ld_sensor1_old, b0, b1, b2, b3, b4, b5,
    b6, b7, b8, '1');
139 ld_lux_area2_sf = f_ad_to_lux(ld_sensor2_old, c0, c1, c2, c3, c4, c5,
    c6, c7, c8, '2');
140 ld_lux_area3_sf = f_ad_to_lux(ld_sensor3_old, d0, d1, d2, d3, d4, d5,
    d6, d7, d8, '3');
141
142 ld_lux_area0_ef = f_ad_to_lux(ld_sensor0_ef, a0, a1, a2, a3, a4, a5, a6
    , a7, a8, '0');
143 ld_lux_area1_ef = f_ad_to_lux(ld_sensor1_ef, b0, b1, b2, b3, b4, b5, b6
    , b7, b8, '1');
144 ld_lux_area2_ef = f_ad_to_lux(ld_sensor2_ef, c0, c1, c2, c3, c4, c5, c6
    , c7, c8, '2');
145 ld_lux_area3_ef = f_ad_to_lux(ld_sensor3_ef, d0, d1, d2, d3, d4, d5, d6
    , d7, d8, '3');
146
147 // Valor de entrada sera o Lux Filtrado
148 entrada_0 = ld_lux_area0_ef;
149 entrada_1 = ld_lux_area1_ef;
150 entrada_2 = ld_lux_area2_ef;
151 entrada_3 = ld_lux_area3_ef;
152
153 PID_Area0.Compute();
154 PID_Area1.Compute();
155 PID_Area2.Compute();
156 PID_Area3.Compute();
157
158 /* Saida PWM */
```

```

159 analogWrite (pwmPin2, saida_0);
160 analogWrite (pwmPin3, saida_1);
161 analogWrite (pwmPin4, saida_2);
162 analogWrite (pwmPin5, saida_3);
163
164 /* Output */
165 MyPlot.SendData("AD_0_sf", Id_lux_area0_sf);
166 MyPlot.SendData("AD_1_sf", Id_lux_area1_sf);
167 MyPlot.SendData("AD_2_sf", Id_lux_area2_sf);
168 MyPlot.SendData("AD_3_sf", Id_lux_area3_sf);
169
170 MyPlot.SendData("AD_0_cf", Id_lux_area0_cf);
171 MyPlot.SendData("AD_1_cf", Id_lux_area1_cf);
172 MyPlot.SendData("AD_2_cf", Id_lux_area2_cf);
173 MyPlot.SendData("AD_3_cf", Id_lux_area3_cf);
174
175 MyPlot.SendData("AD_0_ef", Id_lux_area0_ef);
176 MyPlot.SendData("AD_1_ef", Id_lux_area1_ef);
177 MyPlot.SendData("AD_2_ef", Id_lux_area2_ef);
178 MyPlot.SendData("AD_3_ef", Id_lux_area3_ef);
179
180 MyPlot.SendData("SAIDA_AREA_0(%)", f_percent_saida(saida_0));
181 MyPlot.SendData("SAIDA_AREA_1(%)", f_percent_saida(saida_1));
182 MyPlot.SendData("SAIDA_AREA_2(%)", f_percent_saida(saida_2));
183 MyPlot.SendData("SAIDA_AREA_3(%)", f_percent_saida(saida_3));
184 }
185
186
187 long f_ad_to_lux(double Id_ad_sensor, double a0, double a1, double a2,
double a3, double a4, double a5, double a6, double a7, double a8, char
ls_tipo)
188 {
189     long ll_return = 0;
190
191     if ((ls_tipo == '0' and Id_ad_sensor > 1006) or (ls_tipo == '1' and
Id_ad_sensor > 1020) or (ls_tipo == '2' and Id_ad_sensor > 1003) or
(ls_tipo == '3' and Id_ad_sensor > 1022))
192     {
193         return 0;
194     }
195
196     ll_return = a8*pow(Id_ad_sensor,8) + a7*pow(Id_ad_sensor,7) + a6*pow(
Id_ad_sensor,6) + a5*pow(Id_ad_sensor,5) + a4*pow(Id_ad_sensor,4) +
a3*pow(Id_ad_sensor,3) + a2*pow(Id_ad_sensor,2)+ a1*Id_ad_sensor +
a0;
197
198     return ll_return;

```

```
199 }
200
201 double f_exponential_filter(double ld_ad_sensor, char ls_tipo)
202 {
203     double ll_return = 0;
204
205     if (ls_tipo == '0')
206     {
207         X_t_0 = ld_ad_sensor * 5.0/1023.0;
208
209         S_t_0 = alp * X_t_0 + (1 - alp) * S_t1_0;
210
211         S_t1_0 = S_t_0;
212
213         ll_return = S_t_0;
214     }
215     else if (ls_tipo == '1')
216     {
217         X_t_1 = ld_ad_sensor * 5.0/1023.0;
218
219         S_t_1 = alp * X_t_1 + (1 - alp) * S_t1_1;
220
221         S_t1_1 = S_t_1;
222
223         ll_return = S_t_1;
224     }
225     else if (ls_tipo == '2')
226     {
227         X_t_2 = ld_ad_sensor * 5.0/1023.0;
228
229         S_t_2 = alp * X_t_2 + (1 - alp) * S_t1_2;
230
231         S_t1_2 = S_t_2;
232
233         ll_return = S_t_2;
234     }
235     else
236     {
237         X_t_3 = ld_ad_sensor * 5.0/1023.0;
238
239         S_t_3 = alp * X_t_3 + (1 - alp) * S_t1_3;
240
241         S_t1_3 = S_t_3;
242
243         ll_return = S_t_3;
244     }
245 }
```

```
246
247     ll_return = ll_return *1023.0/5.0;
248
249     return ll_return;
250 }
251
252 long f_percent_saida(double Id_saida)
253 {
254     long ll_pr_uso;
255
256     ll_pr_uso = (Id_saida*100)/255;
257
258     return ll_pr_uso;
259 }
```

Listing 6 – Software Malha Fechada 4 Áreas