

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Carlo dos Santos Pimentel

**MÓDULO DE CARGAS RESISTIVAS COM INTERFACE MICROCONTROLADA
PARA APLICAÇÃO EM ENSAIOS DE MÁQUINAS ELÉTRICAS**

Florianópolis

2021

Carlo dos Santos Pimentel

**MÓDULO DE CARGAS RESISTIVAS COM INTERFACE MICROCONTROLADA
PARA APLICAÇÃO EM ENSAIOS DE MÁQUINAS ELÉTRICAS**

Trabalho de Conclusão do Curso de Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Carlos Alberto Livramento

Coorientador: Cristian Franzoi Mazzola

Florianópolis

2021

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

dos Santos Pimentel, Carlo
MÓDULO DE CARGAS RESISTIVAS COM INTERFACE
MICROCONTROLADA PARA APLICAÇÃO EM ENSAIOS DE MÁQUINAS
ELÉTRICAS / Carlo dos Santos Pimentel ; orientador, Carlos
Alberto Livramento, coorientador, Cristian Franzoi
Mazzola, 2021.
133 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia Elétrica, Florianópolis, 2021.

Inclui referências.

1. Engenharia Elétrica. 2. Ensaio em Máquinas
Elétricas. 3. Módulo de Cargas Resistivas. 4. Lâmpadas
incandescentes como carga resistiva. I. Alberto
Livramento, Carlos. II. Franzoi Mazzola, Cristian . III.
Universidade Federal de Santa Catarina. Graduação em
Engenharia Elétrica. IV. Título.

Carlo dos Santos Pimentel

MÓDULO DE CARGAS RESISTIVAS COM INTERFACE MICROCONTROLADA PARA APLICAÇÃO EM ENSAIOS DE MÁQUINAS ELÉTRICAS

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia Elétrica” e aceito, em sua forma final, pelo Curso de Graduação em Engenharia Elétrica.

Florianópolis, 31 de agosto de 2021.



Documento assinado digitalmente
Jean Viane Leite
Data: 03/09/2021 00:30:03-0300
CPF: 003.474.909-80
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Jean Viane Leite, Dr.
Coordenador do Curso de Graduação em Engenharia Elétrica

Banca examinadora:



Documento assinado digitalmente
Carlos Alberto Livramento
Data: 08/09/2021 11:51:33-0300
CPF: 245.267.709-49
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Carlos Alberto Livramento
Orientador - Universidade Federal de Santa Catarina



Documento assinado digitalmente
Cristian Franzoi Mazzola
Data: 31/08/2021 18:58:23-0300
CPF: 091.191.289-40
Verifique as assinaturas em <https://v.ufsc.br>

Eng. Eletric. Cristian Franzoi Mazzola
Coorientador - Universidade Federal de Santa Catarina



Documento assinado digitalmente
Jean Viane Leite
Data: 03/09/2021 00:29:33-0300
CPF: 003.474.909-80
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Jean Viane Leite, Dr.
Universidade Federal de Santa Catarina



Documento assinado digitalmente
Renato Lucas Pacheco
Data: 31/08/2021 19:03:01-0300
CPF: 341.751.489-49
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Renato Lucas Pacheco, Dr.
Universidade Federal de Santa Catarina



Documento assinado digitalmente
Felipe Henrico Leite Ferraz de Campos
Data: 31/08/2021 21:20:06-0300
CPF: 057.320.209-50
Verifique as assinaturas em <https://v.ufsc.br>

Eng. Eletric. Felipe Henrico Leite Ferraz de Campos
Universidade Federal de Santa Catarina

Esta monografia é dedicada aos meus pais Ari Renato de Lima Pimentel e Rosângela Martins dos Santos. Obrigado por fazerem o possível e impossível por mim, o amor de vocês me inspira.

AGRADECIMENTOS

Neste momento de reflexão dos agradecimentos a aproximadamente 12.000km da minha família e de meus melhores amigos, tive a sensação de que este poderia ser o capítulo mais importante deste trabalho. Portanto, foi com enorme prazer que utilizei um par de horas relembrando, nostalgicamente, minha trajetória até aqui, antes da elaboração do mesmo. E que boas recordações.

Primeiramente, gostaria de agradecer ao meu coorientador e amigo Cristian Franzoi Mazzola por todo apoio durante a elaboração deste trabalho. Aproveito ainda para parabenizá-lo pela sua dedicação e curadoria, por assim dizer, ao Laboratório de Máquinas. Você tem o dom de ensinar.

Agradeço ao meu avô Adão V. Alves dos Santos, que me introduziu aos conceitos de eletricidade, ainda muito novo, com seu vasto conhecimento empírico na área.

Às minha avós, Maria Helena M. dos Santos e Maria Emília Pimentel, que tanto me ensinaram sobre a vida.

Agradeço em especial ao meu amigo Pedro M. Locatelli e sua família por todo apoio durante esse ciclo. Obrigado por tantos ensinamentos e memórias especiais.

Ao Felipe B. Coelho, obrigado pela parceria nos negócios e pelos ensinamentos técnicos na área. Desejo todo sucesso do mundo no seu novo empreendimento.

Ao meu amigo e também sócio Matheus M. Livramento, por todas aventuras e bons momentos durante essa caminhada.

Agradeço, também, ao meu amigo Matheus G. Lintemani, por me hospedar incontáveis vezes em sua casa. Espero poder te recompensar com uma hospedagem por aqui agora.

Ao Danilo F. Bento, obrigado por tornar a caminhada muito mais leve e divertida. Você é exemplo de perseverança.

Ao Felipe B. Soares, por todas conversas e sábias palavras.

Ao grande amigo e vizinho Gabriel Romariz e sua família, por todo apoio de sempre.

Agradeço a todas amizades que fiz durante todo período da graduação. A saudade que eu sinto ao escrever agora me faz lembrar o quão especial foi realizar essa jornada ao lado de vocês.

Agradeço também aos amigos Allanis, Andrei, Bianca, Cisso, Darlan, Diego, Daniel, Felipe, Henrique, Isabela, João, Mario, Marco, Maiara, Pedro e Sérgio, que conheci do outro lado mundo, mas também me apoiaram e incentivaram nessa jornada.

Também quero agradecer ao meus chefes Caroline Olson e Simon Winter por cederem o escritório depois do expediente de trabalho para realização deste TCC.

À Universidade Federal de Santa Catarina, em especial ao corpo docente e funcionários do Departamento de Engenharia Elétrica e Eletrônica, por todo o conhecimento transmitido e serviço prestado.

Por fim, obrigado pai e mãe por me apoiar e me dar forças para as longas jornadas de estudo antes de cada prova. Por não me deixar desanimar com sábias palavras quando os resultados não eram positivos. Por me fazerem acreditar em mim mesmo e no meu potencial. E por me ensinaram tanto através do exemplo de vocês. Muito obrigado.

”Sonhar é acordar-se para dentro“ - Mário Quintana.

RESUMO

Os ensaios e testes em máquinas elétricas com cargas resistivas experimentados em laboratório possuem expressiva importância no ensino de engenharia elétrica. Assim, a infraestrutura dos laboratórios acadêmicos auxilia didaticamente no aprendizado do discente. A proposta deste trabalho consiste em realizar o projeto e execução de um módulo de cargas resistivas, com interface de controle em Arduino, para ensaios realizados em máquinas elétricas no Laboratório de Ensino de Máquinas e Acionamentos Elétricos do Departamento de Engenharia Elétrica e Eletrônica da Universidade Federal de Santa Catarina (LABMAQ-EEL/UFSC). O sistema de comando do projeto utiliza um microcontrolador, módulo LCD 16x2 *keypad shield* e dois módulos de relés com 16 canais cada. O circuito de potência microcontrolado é composto por um sistema trifásico de 6 cargas por fase, sendo cada carga composta por duas lâmpadas incandescentes em série, de 150 W e 127 V nominal. Por meio do teclado e *display* LCD, o operador pode alterar gradualmente o valor da carga resultante por meio de associações em série e em paralelo entre as lâmpadas.

Palavras-chave: Ensaios em Máquinas Elétricas. Módulo de Cargas Resistivas. Lâmpadas incandescentes como carga resistiva.

ABSTRACT

The tests on electrical machines with resistive loads performed in the laboratory have significant importance in the teaching of electrical engineering. Thus, the infrastructure of academic laboratories helps didactically in student learning. The aim of this study consists of carrying out the design and execution of a resistive load module with a control interface in Arduino for tests on electrical machines at the Machines and Electrical Drives Laboratory of the Department of Electrical and Electronic Engineering of the Federal University of Santa Catarina. The project of control system uses a microcontroller, 16x2 LCD *keypad shield* and two 16 channels relay modules. The power circuit consists of a three-phase system of 6 loads per phase, each load being composed of two incandescent lamps in series, of 150W and 127 V nominal. By means of the keyboard and LCD display, the operator can gradually change the value of the resulting load through series and parallel associations between the lamps.

Key-words: Tests on electrical machines. Resistive module panel. Incandescent lamps as resistive load.

LISTA DE ILUSTRAÇÕES

Figura 1 – Sistema Análogo para Triângulo de Potências	20
Figura 2 – Resistor de Fio Enrolado 2kW <i>TE Connectivity</i>	22
Figura 3 – Aumento da Temperatura da Superfície versus Recomendação Percentual de Potência Dissipada pelo Fabricante	23
Figura 4 – Características Construtivas da Lâmpada Incandescente	24
Figura 5 – Filamento de Bobina Enrolada de uma Lâmpada de 60W	25
Figura 6 – Esquema de Ligação para Medição de Tensão e Corrente Lâmpada Incandescente 60W	26
Figura 7 – Gráfico para Lâmpadas Incandescentes A,B e C de Potências Distintas	28
Figura 8 – Dados de Placa Máquina Síncrona LABMAQ-EEL/UFSC	32
Figura 9 – Painel de Cargas Resistivas LABMAQ-EEL/UFSC	33
Figura 10 – Diagrama de Blocos do Sistema do Módulo de Cargas	34
Figura 11 – Circuito Elétrico Trifásico para Chaveamento das Cargas	36
Figura 12 – Microcontrolador Arduino Mega 2560	39
Figura 13 – Relés de Chaveamento para o Circuito Elétrico	39
Figura 14 – LCD <i>Keypad Shield</i> 16x2	40
Figura 15 – Configurações Possíveis para Cargas Trifásicas	44
Figura 16 – Bornes de Conexão para o Painel de Cargas	44
Figura 17 – Interface de Controle para o Usuário	46
Figura 18 – Diagrama Elétrico do <i>datasheet</i> do LCD <i>Keypad Shield</i>	57
Figura 19 – Circuito Elétrico LCD <i>Keypad Shield</i> Projetado no Ambiente ISIS	58
Figura 20 – Simulação Display LCD 16x2 <i>Keypad Shield</i>	58
Figura 21 – Circuito de Potência Completo	59
Figura 22 – Simulação <i>Display</i> e Carga Operando em Circuito Monofásico 220 V/300 W	60
Figura 23 – Simulação Modo Avaliação	61
Figura 24 – Layout Interno do Painel de Cargas	62
Figura 25 – Layout Externo Painel de Cargas	62
Figura 26 – Layout da Peça para o Encaixe do <i>Keypad Shield</i>	63
Figura 27 – Placa <i>Split</i> Para Conexão Arduino-Relés	64
Figura 28 – Implementação Módulo de Cargas	65

LISTA DE TABELAS

Tabela 1 – Valores Nominiais dos Transformadores Ensaaiados no LABMAQ-EEL/UFSC.	31
Tabela 2 – Esquema para Conexão do <i>Keypad Shield</i> nas Portas Lógicas do Arduino	41
Tabela 3 – Circuitos Monofásicos e Trifásicos e seus Respectiveos Valores de Referência de Potência	42
Tabela 4 – Nível Lógico dos Relés para os Distintos Chaveamentos	42
Tabela 5 – Conversão AD dos Botões <i>Keypad Shield</i>	47
Tabela 6 – Esquema para Conexão dos Módulos de Relés nas Portas Lógicas do Arduino	64

LISTA DE QUADROS

Quadro 1 – Filamento de Bobina Enrolada de uma Lâmpada Incandescente de 60W 27

LISTA DE ALGORITMOS

Algoritmo 1 – Funções do <i>Loop</i> Principal no Código Fonte	46
Algoritmo 2 – Lógica de Controle dos Botões Keypad Shield	48
Algoritmo 3 – Lógica de Controle para a Função <i>dispMenu</i>	49
Algoritmo 4 – Lógica de Controle para a Função <i>changeMenu</i>	50
Algoritmo 5 – Lógica de Controle para a Função <i>changeMenu</i>	52
Algoritmo 6 – Lógica de Controle para a Função <i>changeMenu</i>	53
Algoritmo 7 – Lógica de Controle para o Modo Avaliação	54

LISTA DE ABREVIATURAS E SIGLAS

CA	Corrente Alternada
CC	Corrente Contínua
FP	Fator de Potência
S	Potência Aparente
LABMAQ	Laboratório de Ensino de Máquinas e Acionamentos Elétricos
UPS	Uninterruptible Power Supply

LISTA DE SÍMBOLOS

W	Watts
V	Volts
R	Resistência
VA	Potência Aparente
VA_r	Potência Reativa
I_a	Corrente de Armadura
I_f	Corrente de Campo
E_f	Tensão Induzida
Ω	Ohms
ΔT	Diferença de Temperatura
ΔQ	Variação no Calor Sensível
m	Massa
c	Calor Específico do Material

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Motivação	18
1.2	Objetivos	19
1.2.1	Objetivo Geral	19
1.2.2	Objetivos Específicos	19
2	EMPREGO DE CARGAS RESISTIVAS EM ENSAIOS LABORATORIAIS DE MÁQUINAS ELÉTRICAS	20
2.1	Unidades Resistivas de Potência	21
2.2	Uso de Lâmpadas Incandescentes como Carga Resistiva	23
3	ENSAIOS LABORATORIAIS REALIZADOS NO LABMAQ-EEL/UFSC	30
3.1	INTRODUÇÃO	30
3.2	Situação Atual	32
4	PROJETO DO MÓDULO DE CARGAS COM INTERFACE MICROCONTROLADA	34
4.1	Descrição do Sistema de Potência	35
4.1.1	Tipo de Carga	35
4.1.2	Circuito Elétrico e Chaveamento	35
4.2	Descrição Sistema de Comando	37
4.2.1	Sistemas Embarcados	37
4.2.2	Relés de Chaveamento	39
4.2.3	Periférico <i>Keypad Shield 16x2</i>	40
4.2.4	Conexão das Cargas	41
4.2.5	Conexão Carga Trifásica	43
4.2.5.1	Trifásico Estrela 220 V e 380 V	44
4.2.5.2	Trifásica Triângulo 220 V	45
4.2.6	Programação do Código de Controle	45
4.2.6.1	Lógica de Controle da Função <i>KeyboardRead</i>	47
4.2.6.2	Lógica de Controle da Função <i>dispMenu</i> e <i>changeMenu</i>	48
4.2.6.3	Lógica de Controle para o Modo Avaliação	53
5	SIMULAÇÃO COMPUTACIONAL DO SISTEMA DE COMANDO	56
5.1	Simulação Módulo de Cargas	56
5.1.1	Simulação Menu Display Keypad Shield	56
5.1.2	Simulação Circuito Elétrico Monofásico	59

5.1.3	Simulação Circuito Elétrico Trifásico	60
5.1.4	Simulação Modo Avaliação	60
6	IMPLEMENTAÇÃO DO PROJETO	62
7	POSSÍVEIS AJUSTES E SUGESTÕES PARA FUTUROS TRABALHOS	66
	REFERÊNCIAS	67
	APÊNDICE A – CÓDIGO FONTE MÓDULO DE CARGAS	70

1 INTRODUÇÃO

Nos cursos de graduação em áreas tecnológicas, as disciplinas práticas desempenham um papel fundamental no processo de aprendizagem dos discentes. Em conjunto com as aulas teóricas, os experimentos em laboratório trazem uma abordagem técnica da implementação da teoria estudada. Além disso, durante as execuções das atividades propostas nas aulas práticas, há um contato mais próximo entre estudante e professor, que muitas vezes é capaz de proporcionar um ambiente mais aberto a novos questionamentos e aprendizagens sobre o tema (PRESTES; RODRIGUES, 2020).

Para que um laboratório científico possa melhorar o aproveitamento acadêmico dos discentes, é necessário que o mesmo possua boa infraestrutura instrumental. É importante que as ferramentas e máquinas utilizadas nas aulas laboratoriais possuam tecnologia atual, mantendo o estudante atualizado com relação ao mercado de trabalho. A aquisição ou desenvolvimento de novos equipamentos para os laboratórios são de extrema importância, uma vez que um dos papéis da universidade é prover a inovação tecnológica por meio do incentivo à pesquisa. Desta forma, exalta-se ainda mais a importância do laboratório em áreas tecnológicas e se salienta a necessidade de maior atenção quanto à utilização dos recursos disponibilizados em termos de que tipo de equipamento está sendo adquirido. Sendo assim, é importante que os laboratórios possam ser utilizados para a aplicação didática nas disciplinas às quais estão ligados e, ainda, sirvam de apoio para pesquisas e atividades de extensão (PRESTES; RODRIGUES, 2020).

Este trabalho, portanto, visa ao projeto e execução de um módulo de cargas resistivas com interface microcontrolada. O mesmo será utilizado nos ensaios de máquinas elétricas realizados no Laboratório de Ensino de Máquinas e Acionamentos Elétricos do Departamento de Engenharia Elétrica e Eletrônica da Universidade Federal de Santa Catarina (LABMAQ-EEL/UFSC).

1.1 MOTIVAÇÃO

Projetos elaborados por estudantes que visam implementações de melhorias em laboratórios, com foco na eficiência e baixo custo, são essenciais no meio acadêmico. Espera-se que as tecnologias desenvolvidas ofereçam aos laboratórios um ambiente mais agradável, seguro e eficiente para a aprendizagem. Dentre os projetos realizados no LABMAQ-EEL/UFSC, destaca-se o trabalho de conclusão de curso realizado por Gabriel Piazero Hessmann. Publicado em 2018, o trabalho de Hessmann propôs a realização de um sistema automatizado para imposição de faltas em uma máquina síncrona modificada, objetivando, sobretudo, maior praticidade, segurança e eficiência no processo de ensaio (HESSMANN,

2018). Projetos desta natureza, além de serem viáveis economicamente, também se tornam temas para posteriores estudos no meio universitário. Visto a importância dos laboratórios neste meio, o presente trabalho tem sua justificativa assentada na melhoria da infraestrutura instrumental dos ensaios com cargas resistivas em máquinas elétricas presentes no LABMAQ-EEL/UFSC.

1.2 OBJETIVOS

Os objetivos deste trabalho são apresentados na sequência.

1.2.1 Objetivo Geral

Os ensaios com cargas resistivas são realizados no LABMAQ-EEL/UFSC com o finalidade de demonstrar didaticamente aos discentes características específicas das máquinas elétricas. Atualmente, o painel de cargas disponível no laboratório é “montado” e “desmontado” especificamente de acordo com o roteiro de cada ensaio. O módulo de cargas irá substituir o painel de cargas atual por um sistema que busca solucionar os inconvenientes existentes de limitação das cargas, previsibilidade da potência desempenhada e necessidade de ajuste manual das cargas para variar a potência de operação. Sendo assim, o objetivo geral deste trabalho de conclusão de curso consiste em realizar o projeto e a execução de um módulo de cargas resistivas com interface de controle microcontrolada para ensaios realizados em máquinas elétricas no LABMAQ-EEL/UFSC.

1.2.2 Objetivos Específicos

Os objetivos específicos são:

- Analisar o tipo de carga resistiva a ser utilizada no projeto, levando em consideração o balanço entre a viabilidade econômica e a eficiência.
- Estudar os tipos de ensaios laboratoriais realizados com cargas resistivas nas máquinas elétricas presentes no LABMAQ-EEL/UFSC.
- Projetar um circuito elétrico com chaveamento de cargas resistivas que, associadas de maneiras distintas, dissiparão potências elétricas monofásicas e trifásicas predeterminadas.
- Desenvolver um sistema de comando microcontrolado para acionar e controlar os chaveamentos dos distintos valores de potência.
- Realizar a simulação computacional do sistema de comando.
- Implementar o projeto do painel de cargas, validando-o experimentalmente.

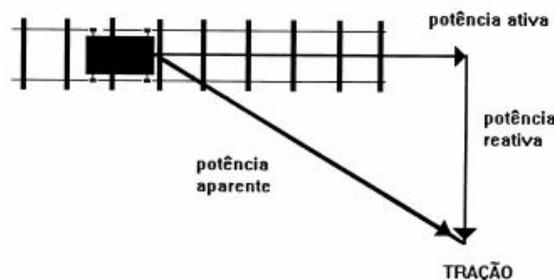
2 EMPREGO DE CARGAS RESISTIVAS EM ENSAIOS LABORATORIAIS DE MÁQUINAS ELÉTRICAS

No sistema elétrico brasileiro, a maior parte das cargas das unidades consumidoras, tais como motores elétricos, transformadores, lâmpadas de descarga, fornos de indução entre outras, consome uma parcela de energia reativa indutiva. Cargas desta natureza necessitam de campo eletromagnético para seu funcionamento e, portanto, sua operação requer potência ativa e reativa (CODI, 2004).

Energia reativa, expressa em quilovolt-ampère-reactivo-hora (kvarh), é a energia elétrica que circula continuamente entre os diversos campos elétricos e magnéticos de um sistema de corrente alternada, sem produzir trabalho, mas é imprescindível para produzir o fluxo magnético necessário ao funcionamento dos motores, transformadores, etc. A energia reativa ocupa espaço no sistema que poderia ser usado por mais energia ativa, portanto, menos energia reativa implica em mais energia ativa disponível no sistema elétrico, reduzindo perdas tanto no sistema quanto nas instalações dos consumidores (QUEIROZ, 2011 apud LIGHT, S/A, p. 44).

Enquanto a potência ativa é sempre consumida na execução de trabalho mecânico, calor, luz ou som, a potência reativa, além de não produzir serviços, circula entre a carga e a fonte de alimentação ocupando “espaço” no sistema elétrico. A potência ativa e a potência reativa, juntas, constituem a potência aparente, medida em kVA (quilo-volt-ampère), que é a potência total gerada e transmitida à carga. A Figura 1 ilustra a potência reativa fazendo uma analogia a um vagão se deslocando sobre os trilhos por ação de uma força não paralela, formando um triângulo de potências. Este, por sua vez, mostra, graficamente, a relação entre as potências ativa, reativa e aparente (CODI, 2004).

Figura 1 – Sistema Análogo para Triângulo de Potências



Fonte: Christo e Souza (2006)

A razão entre a potência ativa e a potência aparente de qualquer instalação se constitui no fator de potência. Este indica qual porcentagem da potência total fornecida

(kVA) é efetivamente utilizada como potência ativa (kW). Portanto, o fator de potência mostra o grau de eficiência do uso dos sistemas elétricos. Valores altos de fator de potência (próximos a 1,0) indicam uso eficiente da energia elétrica, enquanto valores baixos evidenciam seu mau aproveitamento, além de representar uma sobrecarga para todo o sistema elétrico (CODI, 2004).

Em ensaios laboratoriais, para que seja possível operar com fator de potência próximo de 1, buscam-se utilizar cargas resistivas lineares. Assim, torna-se possível obter as curvas de rendimento e de regulação em máquinas elétricas e obter informações importantes, como a eficiência e a queda na tensão de saída, quando operando com carga, por exemplo. Ainda neste âmbito, estes ensaios também são realizados para verificar distintos pontos de operação nas máquinas elétricas.

Motores e transformadores, por exemplo, desenvolvem um campo magnético interno devido à passagem de corrente elétrica nos enrolamentos. Quando estes equipamentos são alimentados em corrente alternada (CA), a energia armazenada em forma de campo magnético tende a se opor à variação da intensidade da corrente, causando um atraso da corrente em relação à tensão. Uma parcela desta corrente não realiza trabalho útil, produzindo, assim, energia reativa (CHRISTO; SOUZA, 2006). Além disso, as reatâncias magnetizantes de cargas indutivas não lineares podem gerar harmônicas no circuito elétrico. Os circuitos elétricos empregados, de modo geral, se utilizam de modelos matemáticos como fasores, para realizar os cálculos e obter seus parâmetros. No entanto, os fasores são definidos para sistemas perfeitamente senoidais, ou seja, sem a presença de harmônicas (POMILIO, 1998).

Diante disso, o uso das cargas indutivas em ensaios laboratoriais de máquinas elétricas demandam maior cautela e análise em relação aos dados a serem obtidos, quando em comparação aos ensaios com cargas resistivas lineares. Em um circuito CA puramente resistivo, as formas de onda de tensão e corrente estão em fase uma com a outra, mudando a polaridade no mesmo instante em cada ciclo e toda a energia entregue à carga é consumida pela mesma. Sendo assim, o uso de cargas não lineares nos ensaios torna-se indesejado, pois, sobretudo, inviabiliza a análise tradicional de circuitos (GRANT, 2019).

2.1 UNIDADES RESISTIVAS DE POTÊNCIA

Resistores são componentes elétricos destinados, em geral, a limitar a intensidade de corrente elétrica. Quando submetido a uma diferença de potencial e uma corrente elétrica estiver circulando em seu interior, o resistor irá transformar toda a energia elétrica em energia térmica. Diante disso, ainda que o mesmo seja utilizado principalmente na limitação de corrente elétrica, existem aplicações onde o foco é a própria dissipação de potência.

Os resistores podem ser, principalmente, de carvão depositado para baixas potências de dissipação e de fio enrolado. O resistor de fio enrolado é comumente utilizado quando há necessidade de maior dissipação de potência. Estes possuem revestimento, geralmente, de esmalte de vidro de cor verde (característico deste tipo de componente), com um núcleo de material isolante, onde é enrolado um fio condutor de resistividade elétrica relativamente alta, conforme ilustra a Figura 2. O material empregado na fabricação do fio de resistência é normalmente o tungstênio, a manganina ou o nicromo e o núcleo isolante é feito de porcelana, baquelita, papel sulfite ou argila cerâmica.

Figura 2 – Resistor de Fio Enrolado 2kW *TE Connectivity*



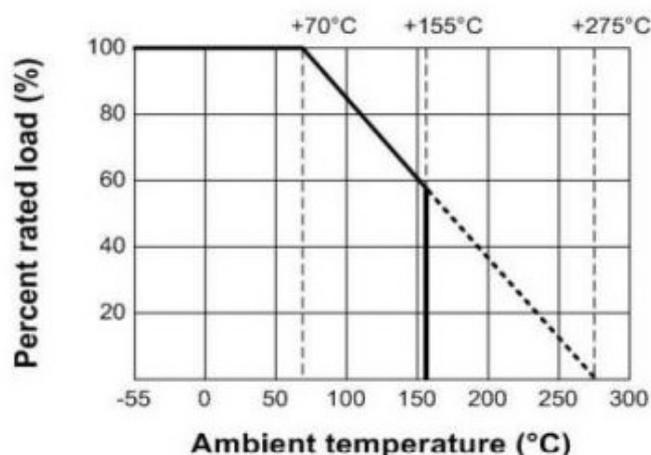
Fonte: [Type TE Series](#) (2018)

De acordo com [Halliday, Resnick e Walker \(2011, p. 192\)](#), “Calor é a energia trocada entre um sistema e o ambiente devido a uma diferença de temperatura”. No caso dos resistores de fio enrolado, a corrente elétrica que circula no enrolamento interno durante sua operação eleva a temperatura do componente. Tendo em vista que o fluido no qual o resistor está em contato é o ar, a troca de energia entre o componente e o ambiente ocorre por convecção. Nesta, a temperatura do ar em contato com resistor aumenta e o ar mais frio e, portanto, mais denso, escoar para tomar o lugar do mais quente, que sobe. Essa diferença do calor cedido pelo resistor e recebido pelo ar é chamada de ΔQ e é calculada através de (2.1).

$$\Delta Q = m.c.\Delta T \quad (2.1)$$

Tendo em vista que a temperatura de operação do resistor de fio enrolado influencia diretamente no seu desempenho, devido a variação de resistividade do enrolamento interno, os fabricantes disponibilizam no *datasheet* do componente as informações de comportamento do mesmo em diferentes temperaturas. O gráfico da Figura 3, retirado do *datasheet* do resistor de fio enrolado de 2kW da marca *TE CONNECTIVITY*, estima as faixas de temperatura e as respectivas percentagens de potência dissipada. Em suma, ao elevar a temperatura de operação, o resistor opera dissipando menos potência. Portanto, para o resistor de fio enrolado, a diferença entre a temperatura ambiente e o componente deve ser tal que o ΔT se mantenha constante na temperatura indicada pelo fabricante.

Figura 3 – Aumento da Temperatura da Superfície versus Recomendação Percentual de Potência Dissipada pelo Fabricante



Fonte: Type TE Series (2018)

Este tipo de resistor é largamente utilizado em bancos de cargas para testes e ensaios em equipamentos de alimentação ininterrupta (UPS). Estes equipamentos fazem parte de um sistema de alimentação de energia elétrica secundário, que, entra em ação, suprindo energeticamente os dispositivos conectados ao mesmo, quando há interrupção no fornecimento de energia elétrica da fonte primária (ELECTRIC, 2019). De forma alternativa, equipamentos desta classe também podem ser testados componente por componente, visando baratear o custo de testes por não utilizarem os bancos de cargas. Todavia, para garantir o correto funcionamento de máquinas elétricas desta natureza, faz-se necessário o teste integral do sistema, utilizando-se equipamentos que permitam simular cenários reais de operação.

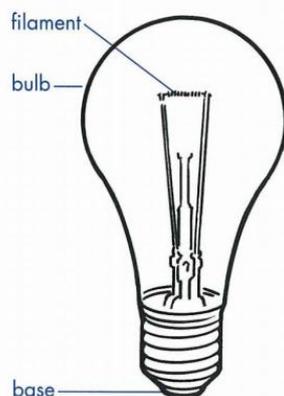
Os testes com carga validam o desempenho operacional de sistemas elétricos e mecânicos destas máquinas, garantindo o correto funcionamento para o uso das mesmas ou como forma de manutenção preventiva. No caso dos geradores reservas, geralmente acionados por motores a diesel para este tipo de aplicação, os testes e ensaios com bancos de cargas revelam problemas em uma situação controlada, antes de uma queda de energia real, aumentando a confiabilidade de sua operação e minimizando os riscos associados a uma falha de energia inesperada (COMRENT, 2019).

2.2 USO DE LÂMPADAS INCANDESCENTES COMO CARGA RESISTIVA

Os resistores de fio enrolado, devido às suas características citadas anteriormente, são uma boa opção de unidade resistiva para projetos onde se deseja ensaiar diferentes pontos de operação em máquinas elétricas. Porém, componentes resistivos que suportam temperaturas elevadas quando em operação tornam-se uma alternativa para este tipo de

aplicação. Um exemplo é a lâmpada incandescente, a qual pode operar em temperaturas elevadas, dissipando potência em forma de calor e luminosidade (SILVA, 2016).

Figura 4 – Características Construtivas da Lâmpada Incandescente



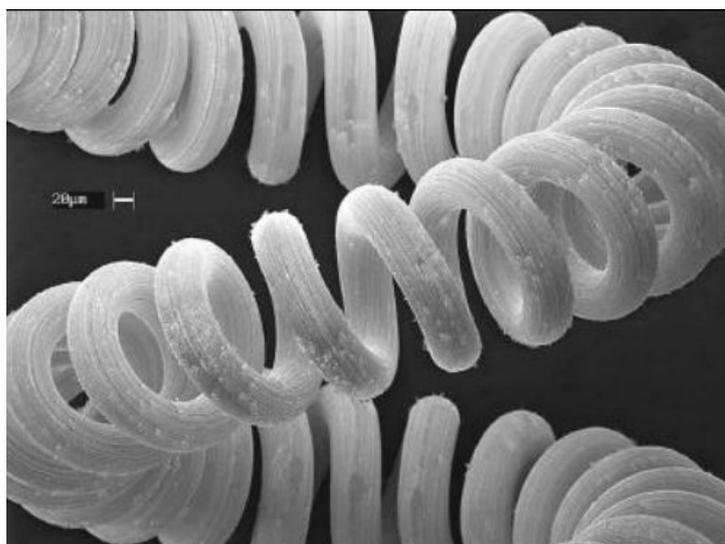
Fonte: Leslie e Conway (1996)

Os estudos que envolvem a criação da lâmpada incandescente iniciaram em meados do ano de 1800 e passaram por diversos ajustes e modificações de inventores ao longo do tempo. Em 1878, Thomas Edison e seu time focaram seus esforços no desenvolvimento do filamento que compõe a lâmpada incandescente. Tal filamento é o responsável por emitir luz quando aquecido pela passagem de corrente elétrica. Nos primeiros testes, foi utilizado um filamento carbonizado de fio de algodão com vida útil de 14,5 horas. Dois anos mais tarde, no início de 1880, Thomas Edison e sua equipe desenvolveram um filamento de bambu carbonizado com vida útil de até 1200 horas. Este filamento se tornou o padrão para Edison e foi utilizado por mais de 10 anos, até que em 1904 os alemães Werner von Bolton e Otto Feuerlien utilizaram o metal tungstênio como filamento para lâmpada (ETHW, 2015) (MATULKA; WOOD, 2013). Este novo filamento apresentou maior vida útil e iluminava mais, quando comparado com o filamento de carbono. Após estas descobertas e melhorias, um grande problema continuou a atormentar Edison e os inventores subsequentes, no que ficou conhecido como “*lamp blackening*” ou “escurecimento da lâmpada”. Isto ocorria porque, ao aquecer o filamento, as moléculas de tungstênio evaporavam e se depositavam na parede interna do invólucro de vidro. Conforme os depósitos engrossavam, a transmissão de luz diminuía até o ponto em que a lâmpada tornava-se inútil. Em 1913, o físico-químico norte americano Irving Langmuir descobriu que, colocando um gás inerte como nitrogênio dentro do bulbo, dobrava a eficiência luminosa da lâmpada (maior luminosidade para a mesma potência aplicada) e diminuía o efeito de escurecimento. Cientistas continuaram desenvolvendo melhorias, reduzindo custos e aumentando a eficiência da lâmpada incandescente nos anos seguintes. Contudo, nenhuma mudança significativa foi implementada desde as lâmpadas incandescentes produzidas a partir de 1950, sendo estas constituídas basicamente de uma ampola de vidro fina, à vácuo

ou gás inerte, um filamento de tungstênio trefilado e uma base condutora (INSTITUTION, 2020) (FURFARI, 2001).

Através da espectroscopia, estudo da interação entre a matéria e a radiação eletromagnética, sabe-se que, para emitir luz visível, um sólido deve ser aquecido acima de 567 °C. Dentre todos materiais de filamento sólido, o tungstênio é o que apresenta a temperatura de fusão mais elevada (3406 °C) e a menor pressão de vapor dos metais puros. A produção da liga de tungstênio utilizada na lâmpada incandescente envolve dopagem por potássio, prensagem, lubrificação, entre outros processos, até chegar na forma espiral final, conforme ilustra a Figura 5. Após estes procedimentos, o filamento de tungstênio, no interior da ampola de vidro, pode operar em temperaturas em torno dos 3000 °C, com uma vida útil de aproximadamente 1000 horas. Os filamentos de baixa temperatura operam na faixa de 2200 a 2400 °C e emitem principalmente energia espectral vermelha. Já os de alta temperatura trabalham na faixa de 2500 a 2600 °C, emitindo mais comprimentos de ondas azuis, sendo paradoxalmente, chamados de luz fria no sentido das cores (MACISAAC; KANNER; ANDERSON, 1999).

Figura 5 – Filamento de Bobina Enrolada de uma Lâmpada de 60W

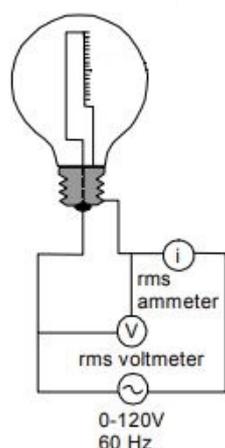


Fonte: Adaptado de MacIsaac, Kanner e Anderson (1999)

Conforme Agrawal, Leff e Menon (1996, p. 649, tradução nossa) “A eficiência da emissão de luz de uma lâmpada incandescente em função da temperatura do filamento é calculada através da teoria de radiação de Planck [...]”. Esta é definida como a razão entre a potência emitida na parte visível do espectro e a potência total de entrada fornecida. Para a lâmpada incandescente operando com temperatura entre 2200 e 2800 °C, a eficiência é na faixa 2% a 13%, ou seja, somente uma pequena parcela da potência fornecida é convertida em luz visível. Também, partindo da teoria de radiação de Planck, a eficácia é definida como a relação entre a saída de luz de uma lâmpada e a energia elétrica

que ela consome. Esta medida, calculada em lumen por watt ($L.W^{-1}$), que consta em praticamente todas as embalagens de lâmpadas, apresenta valores na faixa de 8 a 18 $L.W^{-1}$ para lâmpadas incandescentes. A lâmpada halógena, que possui praticamente a mesma tecnologia da incandescente, conforme será estudado posteriormente, possui eficácia entre 18 e 29 $L.W^{-1}$, por exemplo. Além disso, na medida em que o filamento de tungstênio aquece, sua resistividade aumenta e, conseqüentemente, o valor de resistência também varia. A Figura 6 mostra o esquema para medição de tensão e corrente de uma lâmpada incandescente de 60W através das quais se obtém, analiticamente, os valores de resistência pela divisão da tensão pela corrente. Dos valores medidos, obteve-se o Quadro 1, no qual pode-se observar que o valor de resistência do filamento de tungstênio aumenta conforme este é aquecido (MACISAAC; KANNER; ANDERSON, 1999).

Figura 6 – Esquema de Ligação para Medição de Tensão e Corrente Lâmpada Incandescente 60W



Fonte: Adaptado de MacIsaac, Kanner e Anderson (1999)

O gráfico da Figura 7a mostra o valor da potência elétrica dissipada em função da tensão aplicada para três lâmpadas incandescentes: A, B e C, todas com valores nominais de tensão iguais a 127 V e potências distintas. Para uma tensão de 130 V, as potências consumidas pelas lâmpadas A, B e C são, respectivamente 234 W, 154 W e 116 W. Quando a tensão é reduzida para 65 V, as potências efetivamente medidas são, respectivamente, 80 W, 52 W e 39 W. Neste caso, supondo-se que a resistência não sofresse variação diante diferentes tensões e tendo em vista que a carga é resistiva pura ($FP = 1$), torna-se possível o cálculo da potência para diferentes tensões (2.2).

$$P = \frac{V^2}{R} \quad (2.2)$$

Assim, quando a tensão fosse metade da nominal (65V), os valores esperados para as potências seriam 59 W, 39 W e 29 W. A diferença percentual entre as potências observadas

no gráfico e as calculadas (supondo-se resistência constante) é de 35%. Em complemento, o gráfico da Figura 7b apresenta para as mesmas lâmpadas A, B e C, que o valor da resistência se eleva quando a tensão aplicada aumenta (SILVEIRA, 2015).

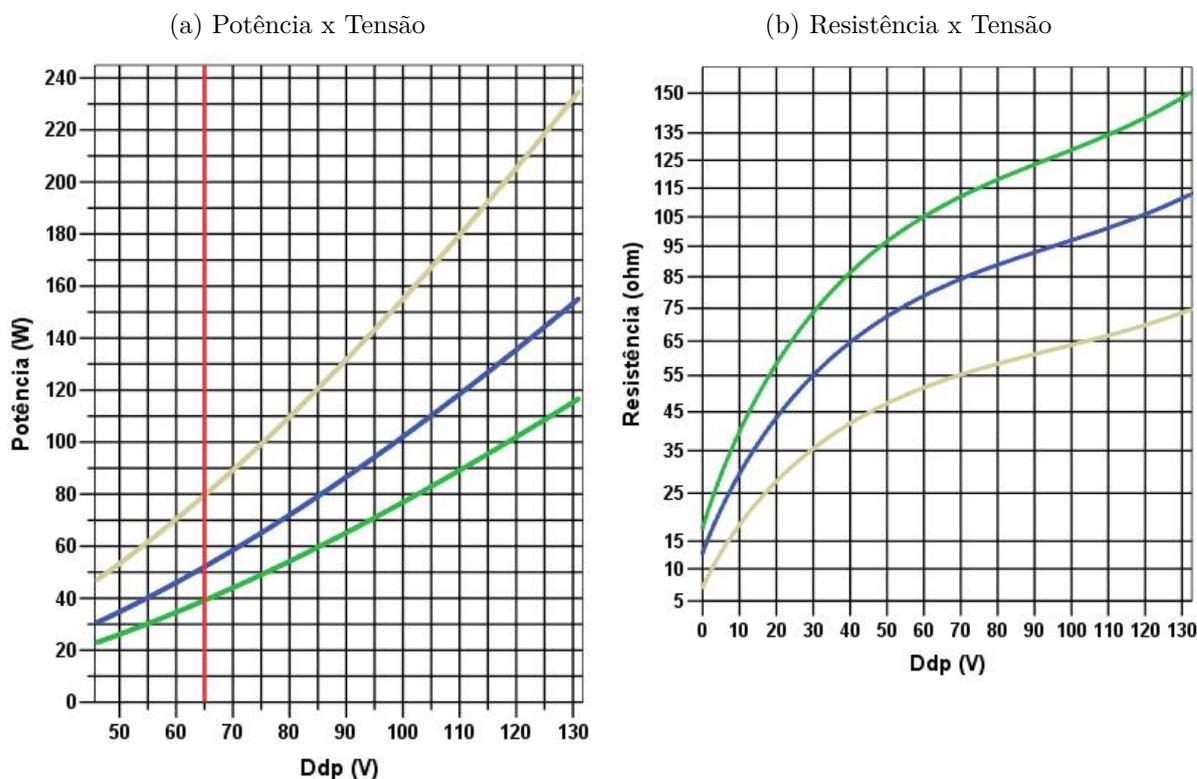
Quadro 1 – Filamento de Bobina Enrolada de uma Lâmpada Incandescente de 60W

rms voltage (volts)	rms current (amperes)	Power (watts)	Filament resistance (ohms)
0	0	0	16.85
70.27	0.3710	26.07	189.4
79.99	0.3976	31.80	201.2
90.00	0.4234	38.11	212.6
100.04	0.4479	44.81	223.3
109.97	0.4710	51.80	233.5
119.95	0.4931	59.15	243.3

Fonte: Adaptado de MacIsaac, Kanner e Anderson (1999)

A relação entre a temperatura e a resistividade dos materiais é bem definida na literatura. Por outro lado, não há tanta clareza na relação entre tensão aplicada e a temperatura. Para o resistor de fio enrolado, estudado anteriormente, a temperatura ambiente influencia diretamente o seu desempenho pelo fato do mesmo operar numa faixa de temperatura baixa. Dessa forma, a variação na resistência do resistor de fio enrolado é pequena. Já a lâmpada incandescente, por operar em temperatura elevada, não é influenciada diretamente pela temperatura ambiente. Todavia, o valor da resistência do filamento de tungstênio quando lâmpada incandescente está desligada é muito menor do que quando em operação.

Figura 7 – Gráfico para Lâmpadas Incandescentes A,B e C de Potências Distintas



Fonte: Adaptado de [Silveira \(2015\)](#)

Devido principalmente à baixa eficiência na emissão de luz das lâmpadas incandescentes, uma tendência mundial se instituiu no início dos anos 2000 visando substituir e, posteriormente, eliminar as lâmpadas incandescentes do mercado. No Brasil, a substituição das lâmpadas incandescentes começou em 2012, com a proibição da venda de lâmpadas com potências superiores a 150W. Entre 2013 e 2016, houve a eliminação das lâmpadas de potências entre 25 W e 100 W. Também, em 2016, foi proibida a fabricação, importação e comercialização deste tipo de lâmpada ([GANDRA, 2016](#)). Embora encontrar lâmpadas incandescentes, principalmente as de maior potência, tenha se tornado uma tarefa difícil, lâmpadas halógenas podem ser encontradas mais facilmente. Tais lâmpadas possuem as características construtivas bem semelhantes as das incandescentes, sendo a principal diferença o gás halogênio em contato com o filamento de tungstênio. Do mesmo modo que ocorre com as lâmpadas incandescentes, os átomos de tungstênio evaporam do filamento quente e se movem em direção à parede mais fria do bulbo. Porém, na lâmpada halógena, átomos de tungstênio, oxigênio e halogênio se combinam na parede do bulbo para formar moléculas de halogenetos de tungstênio. A temperatura da parede do bulbo mantém as moléculas de halogenetos de tungstênio no estado gasoso. As moléculas se movem em direção ao filamento quente, onde a temperatura mais alta as separa. Os átomos de tungstênio são depositados novamente nas regiões mais frias do filamento (não nos lugares

exatos de onde evaporaram) e o ciclo volta a se repetir. Esta sequência é conhecida como ciclo halógeno das lâmpadas incandescentes e não permite que as moléculas de tungstênio se depositem no interior da ampola de vidro, o que evita o escurecimento (FURFARI, 2001).

Embora o valor da resistência da lâmpada incandescente com filamento de tungstênio varie com a temperatura, esta é utilizada largamente em experiências laboratoriais no meio acadêmico (AGRAWAL; LEFF; MENON, 1996). O uso deste componente no presente projeto se justifica principalmente por sere excelente dissipador de potência. Na prática, as lâmpadas não irão trabalhar em sua capacidade nominal durante todo tempo, otimizando a vida útil das mesmas. Além disso, o custo das lâmpadas desta natureza é mais acessível quando em comparação com o custo dos resistores de fio enrolado citados anteriormente. Há ainda outro fator interessante, que é a “visualização do aquecimento”, ou seja, a observação da intensidade da luz emitida e a identificação visual da fonte de calor, evitando queimaduras acidentais.

3 ENSAIOS LABORATORIAIS REALIZADOS NO LABMAQ-EEL/UFSC

3.1 INTRODUÇÃO

Conforme Toro (1994, p. 70), “[...] transformadores de potência e distribuição são projetados para operar sob condições de frequência e tensão eficazes constante. Para estes transformadores os ensaios de rendimento e regulação possuem importância significativa”.

O rendimento de um transformador é definido como a razão entre a potência ativa entregue à carga e a potência ativa de entrada. Em suma, a potência ativa de entrada deverá suprir a potência ativa consumida na carga mais a potência ativa das perdas internas do transformador. O ensaio de rendimento visa obter os valores das potências ativas na entrada e na carga.

Neste trabalho, a regulação de tensão é definida como a razão da diferença entre o valor eficaz da tensão em vazio e o valor eficaz da tensão com carga, e esta tensão eficaz com carga. A regulação nominal é definida como a diferença entre o valor eficaz da tensão em vazio e o valor eficaz da tensão a plena carga, e esta tensão eficaz a plena carga. Dá uma ideia do desempenho de um transformador no que se refere ao grau de constância da tensão de saída quando há variação da carga. O ensaio de regulação visa obter os valores eficazes dessas tensões.

Os ensaios de rendimento e regulação realizados nas aulas práticas no LABMAQ-EEL/UFSC são feitos para transformadores e autotransformadores monofásicos ou trifásicos. Os valores nominais de alguns equipamentos disponíveis estão descritos na Tabela 1.

Para os ensaios deste trabalho, será usado um Varivolt, que é um autotransformador variável, o que permite a energização gradual através da variação do nível de tensão do secundário. No experimento usado como teste, o Varivolt é conectado no lado de baixa tensão do equipamento em teste, enquanto o painel de cargas é conectado no lado de alta tensão. O ensaio consiste em variar a carga desde zero até a potência nominal do transformador, mantendo a tensão nominal constante na carga e anotando os valores de tensão, corrente e potência ativa nos dois lados do transformador. Sugere-se, para se ter uma visão clara da característica de funcionamento do equipamento, a medição de pelo menos seis ou oito valores de cargas com fator de potência unitário (puramente resistivas). Posteriormente, são realizados os cálculos de rendimento e regulação para cada nível de carga medido. Por fim, de posse dos dados e cálculos obtidos através dos ensaios, são determinadas duas curvas características: Rendimento x Potência e Corrente na Carga x Potência.

Tabela 1 – Valores Nominais dos Transformadores Ensaaiados no LABMAQ-EEL/UFSC.

Máquina Elétrica	Tensão (V)	Corrente (A)	Potência (VA)
Transformador monofásico	110/220	3,0/1,5	330
Autotransformador monofásico	330/110	1,5/4,5	495
Autotransformador monofásico	330/220	3,0/4,5	990
Transformador trifásico Y- Δ	380/220	1,5/2,6	990

Fonte: Pimentel (2021)

O ensaio de regulação também é realizado em máquinas síncronas. A regulação mostra a queda de tensão interna, em valores eficazes, da máquina em um dado ponto de operação. É a tensão interna do equipamento menos a tensão de saída, com a diferença dividida pela tensão de saída.

No ensaio de regulação de tensão da máquina síncrona realizado no LABMAQ-EEL/UFSC, inicia-se conectando cargas resistivas com baixa potência e se aumenta gradativamente o valor da carga até ser atingido o valor de potência nominal da máquina. Os valores nominais da máquina usada para validar o estudo deste trabalho de conclusão de curso são mostrados na foto da placa de dados apresentada na Figura 8. No ensaio, a cada acréscimo de carga, é medido o valor da tensão terminal e o valor da corrente de campo correspondente. Sugere-se seis ou mais pontos para que se tenha uma real ideia do comportamento da máquina em estudo.

No ensaio de regulação de tensão da máquina síncrona realizado no LABMAQ-EEL/UFSC, inicia-se conectando cargas resistivas com baixa potência e se aumenta gradativamente o valor da carga até ser atingido o valor de potência nominal da máquina. Os valores nominais da máquina usada para validar o estudo deste trabalho de conclusão de curso são mostrados na foto da placa de dados apresentada na Figura 8. No ensaio, a cada acréscimo de carga, é medido o valor da tensão terminal e o valor da corrente de campo correspondente. Sugere-se seis ou mais pontos para que se tenha uma real ideia do comportamento da máquina em estudo.

Figura 8 – Dados de Placa Máquina Síncrona LABMAQ-EEL/UFSC



Fonte: Arquivo Pessoal

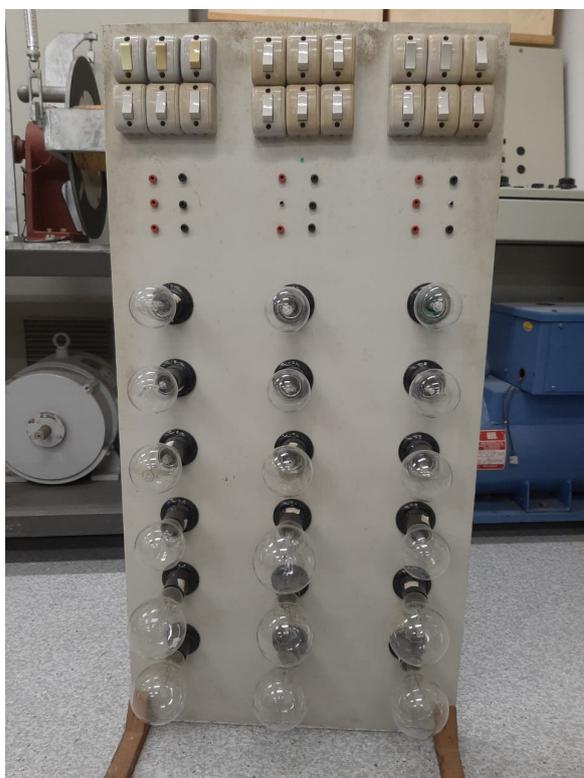
3.2 SITUAÇÃO ATUAL

A carga resistiva atual, empregada nos ensaios, tanto para os transformadores, quanto para os geradores síncronos, é composta por um mesmo conjunto de lâmpadas incandescentes. O dispositivo em questão trata-se de um painel retangular de madeira compensada, com dimensões de aproximadamente 120 cm x 40 cm. Este, possui três fileiras verticais, cada uma contendo seis receptáculos para lâmpadas com conexão do tipo E-27, conforme mostra a fotografia da Figura 9. Cada receptáculo é acionado independentemente pelo seu respectivo interruptor, resultando num total de 18 receptáculos e 18 interruptores. Para cada fileira vertical, o painel dispõem de pares de bornes para conexão da fonte de tensão que irá alimentar as lâmpadas. Para compor a carga, estão disponíveis lâmpadas halógenas de 70 W e incandescentes de 100 e 200 W de potência. O painel é usado para simular cargas monofásicas ou trifásicas, ligadas em estrela ou delta. As aulas laboratoriais são executadas em três grupos de estudantes e, portanto, até a execução deste trabalho, o LABMAQ-EEL/UFSC possuía três painéis de cargas idênticos. Desta forma, cada grupo utiliza de um painel para realização dos ensaios.

A realização dos ensaios com este sistema rudimentar de imposição de cargas apresenta certas dificuldades operacionais. Devido aos distintos ensaios com cargas resistivas realizados no LABMAQ-EEL/UFSC há a necessidade de acrescentar ou retirar as lâmpadas do painel manualmente. Esta tarefa exige cautela, pois as lâmpadas operam em

Figura 9 – Painel de Cargas Resistivas LABMAQ-EEL/UFSC

(a) Disposição Frontal das Cargas



(b) Conexão das Cargas na Parte Traseira



Fonte: Arquivo Pessoal

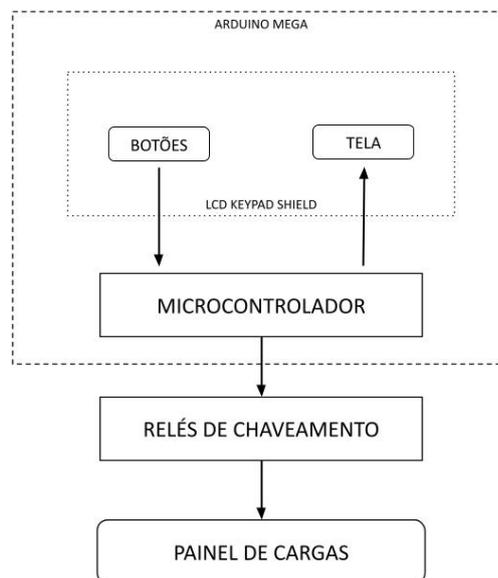
temperaturas elevadas e são frágeis. Por ser composto por lâmpadas de 70, 100 e 200 W, as cargas discretas são muito grandes quando se ensaia um transformador monofásico de 330 VA, por exemplo, e muito pequenas para um gerador de 5 KVA, por exemplo. Além disto, mesmo se todas as lâmpadas fossem de 200 W, não seria possível alcançar a potência nominal do gerador.

4 PROJETO DO MÓDULO DE CARGAS COM INTERFACE MICROCONTROLADA

Ao longo das últimas décadas, o rápido desenvolvimento tecnológico dos microcontroladores e do processamento de dados computacional possibilitou que processos produtivos fossem automatizados. MARCAL, GUIMARAES e RESENDE (2013, p. 4) definem a automação “[...] como a utilização de dispositivos mecânicos ou elétricos para auxiliar e executar algumas funções e visam melhorar a eficiência e a segurança na operação”. Vale salientar que automação industrial difere-se da automação em contexto geral, pois a primeira busca, principalmente, o aumento na eficiência da produção dos insumos (MARAFON et al., 2018).

Neste contexto surgem os sistemas automatizados que podem ser divididos em diferentes áreas de aplicação. O presente trabalho segue basicamente os mesmos passos de desenvolvimento de projetos de Sistemas Industriais de Produção Automatizados, com o diagrama de blocos da Figura 10, ilustrando o processo lógico de funcionamento do módulo de cargas.

Figura 10 – Diagrama de Blocos do Sistema do Módulo de Cargas



Fonte: Arquivo Pessoal

4.1 DESCRIÇÃO DO SISTEMA DE POTÊNCIA

Para melhor compreensão do sistema automatizado do módulo de cargas, o mesmo será estudado a seguir em duas partes principais, sendo elas: Sistema de Potência e Sistema de Comando.

4.1.1 Tipo de Carga

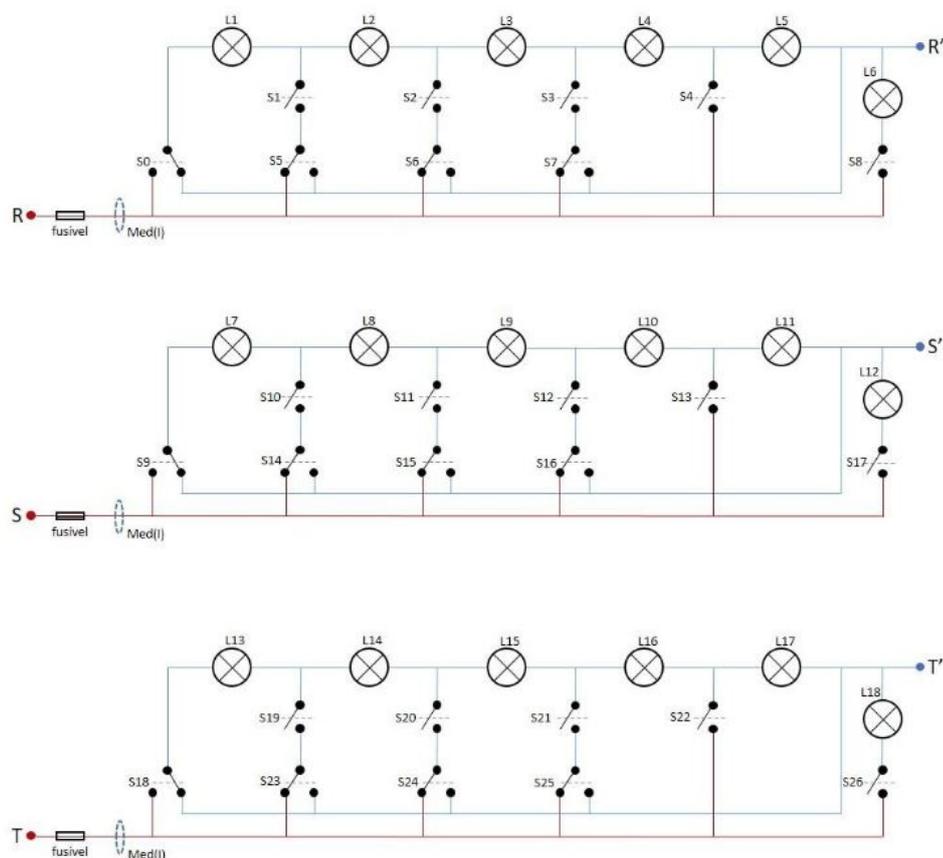
A carga escolhida para o módulo do projeto foi composta por lâmpadas incandescentes com potência e tensão nominais de, respectivamente, 150 W e 127 V. Estas, quando projetadas corretamente num circuito elétrico, podem substituir as unidades resistivas utilizadas comercialmente com uma grande vantagem econômica. Apesar de o filamento de tungstênio das lâmpadas incandescentes operar em temperatura elevada, não há a mesma necessidade de um sistema de resfriamento quando em comparação aos resistores de fio enrolado. A dissipação de calor fica facilitada pela geometria da lâmpada. Por outro lado, há variação na resistividade do filamento da lâmpada incandescente para diferentes níveis de temperatura. Porém, a vantagem econômica, aliada ao fato destes tais tipos de lâmpadas serem cargas resistivas puras ($FP = 1$), as tornam a opção mais adequada ao projeto.

Conforme visto no Capítulo 2, as lâmpadas incandescentes em questão não são mais comercializadas no mercado brasileiro desde 2016. Porém, o LABMAQ-EEL/UFSC conta com um estoque dessas lâmpadas, que foram adquiridas no passado, antes que a lei de 2016 entrasse em vigor. São duráveis e podem suprir o laboratório por um longo tempo.

4.1.2 Circuito Elétrico e Chaveamento

Para o circuito elétrico, objetivou-se principalmente o acionamento das cargas do módulo com o balanço mais eficiente entre o mínimo de chaves (relés) e maior número de associações de cargas possíveis. Isto é, através das conexões série/paralelo entre as lâmpadas, obtém-se distintos valores de resistências equivalentes e, conseqüentemente, diferentes valores de potência dissipada. A Figura 11 mostra a configuração do circuito elétrico chaveado para as três fases, com cargas de L1 até L18, composta de duas lâmpadas incandescentes (150 W/127 V) conectadas em série. Na condição de projeto, optou-se por uma tensão plena de até 220 V, o que permite ensaios até 380 V em sistema trifásico. Assim, cada lâmpada receberá aproximadamente 110 V quando, na verdade, o seu valor nominal é de 127V. Dessa forma, esta diminuição de tensão resultará numa menor potência desenvolvida pela lâmpada, mas, por outro lado, proporcionará o aumento de sua vida útil.

Figura 11 – Circuito Elétrico Trifásico para Chaveamento das Cargas



Fonte: Arquivo Pessoal

Com o número máximo de seis cargas por fase, obteve-se o melhor custo-benefício para o projeto utilizando nove chaves para cada fase. Neste ponto, faz-se importante ressaltar que objetivou-se obter valores de potências dissipadas de forma que, a cada novo chaveamento, houvesse acréscimos gradativos no valor da mesma e que seus intervalos possibilitassem um aumento em rampa. Mais valores de potências intermediários seriam possíveis com adição de chaves ao circuito, porém viu-se desnecessário, tendo em vista que 22 valores distintos de potências são suficientes para obter resultados confiáveis para ensaios com cargas resistivas do LABMAQ-EEL/UFSC. Os nove relés, responsáveis pelas diferentes associações série/paralelo, tornaram possível não somente conectar todas as cargas em série ou paralelo, mas também circuitos com uma única carga atuando, por exemplo. Nesta lógica, projetou-se, então, todos os demais circuitos responsáveis por dissipar as potências intermediárias.

O módulo de cargas opera com seu valor máximo de potência dissipada quando as seis cargas de cada fase estão associadas em paralelo. Neste caso, a potência de referência apresentada ao operador foi de 1800 W, para o circuito monofásico de 220 V, e 5400 W, para os sistemas trifásicos. Todavia, devido ao fato da tensão aplicada às lâmpadas ser de

110 V e não a nominal de 127 V, a potência máxima dissipada foi menor. Para circuitos com resistência constante, uma queda no nível de tensão naturalmente ocasionaria uma diminuição quadraticamente proporcional da potência dissipada. Contudo, conforme é possível observar no gráfico da Figura 7b, a resistência das lâmpadas é alterada em função da tensão e, portanto, essa relação quadrática não pode ser aplicada.

Os valores de referência de potência foram calculados utilizando-se o circuito monofásico de 220 V como referência e, com auxílio de (4.1), derivado de (2.2).

$$P = \frac{V^2}{R} \quad (4.1)$$

Assim, para a carga L1 (par de lâmpadas) operando em condição nominal (300 W), por meio de (4.1), temos uma resistência $R = 161,33 \Omega$. A partir deste resultado, calculou-se os valores de resistência equivalente para cada configuração específica de chaveamento e suas respectivas potências dissipadas. É importante ressaltar que o valor de resistência utilizada para estes cálculos é uma estimativa do valor real, tendo em vista que, conforme o filamento de tungstênio da lâmpada incandescente é aquecido, há variação da sua resistividade. Além disso, as lâmpadas receberão menos tensão que a nominal. Os valores de referência servem apenas para dar uma ideia de um valor que não será ultrapassado na prática. A exata potência dissipada é medida pelo discente. Porém, o mesmo precisa ter um limite máximo de referência para não sobrecarregar o sistema ou queimar os instrumentos, por extrapolar os limites das escalas de medição.

4.2 DESCRIÇÃO SISTEMA DE COMANDO

Seguindo a lógica de automação de projetos, será apresentado nesta seção o sistema de controle no que se refere ao *software* e aos *hardwares* utilizados.

4.2.1 Sistemas Embarcados

Os sistemas embarcados tornaram-se populares por terem capacidade de controlar outros dispositivos físicos e de receber e tratar dados através de uma interface. Por realizarem tarefas de forma limitada, permitiu-se o desenvolvimento de circuitos menores, focados em necessidades específicas e com custo de produção significativamente menor. No princípio, devido ao limitado recurso computacional dos dispositivos embarcados, o usuário ou utilizador possuía pouca ou até mesmo nenhuma interação com o mesmo. Entretanto, com o rápido desenvolvimento computacional com o passar do anos, foi possível uma maior interação do usuário através de interfaces e periféricos (SILVA; ARAUJO; CAVALCANTE, 2019).

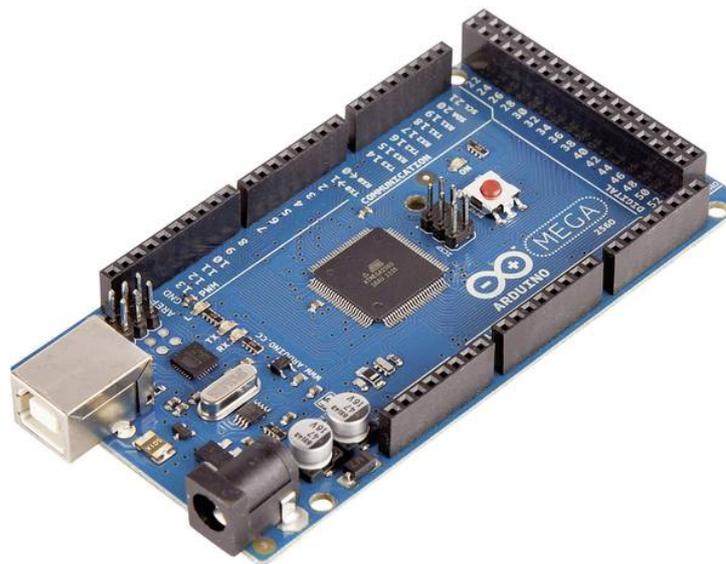
Para McRoberts (2011, p. 3, tradução nossa), “Arduino é um sistema embarcado, ou seja, que pode interagir com seu ambiente por *hardware* e *software* incorporados a

um dispositivo com um objetivo pré-definido”. O Arduino surgiu como uma solução de desenvolvimento e prototipagem, com aplicações que vão desde sistemas de fazer um LED alterar de cor, a projetos complexos que demandam domínio de lógica de programação e eletrônica. Conhecidos também como microcontroladores, podem ser considerados como computadores de tamanho reduzido, desenvolvidos em um único circuito integrado. Dentre os Arduinos mais conhecidos e utilizados, desde o meio acadêmico até entusiastas da área, estão as placas Arduino UNO, Arduino MEGA e Arduino NANO. Estes estão no topo da lista, devido principalmente à grande quantidade de material didático disponível e ao preço acessível (SILVA; ARAUJO; CAVALCANTE, 2019).

Atualmente, o Arduino Uno é a versão mais popular dentre os dispositivos disponíveis da marca e, por isso, existe, na internet, uma infinidade de projetos que o envolve. Entretanto, devido à necessidade de uma quantidade maior de portas lógicas I/O, o Arduino MEGA 2560 se enquadrou melhor para o projeto proposto neste trabalho. Dentre os modelos citados, é o que possui maior capacidade de memória e, devido à quantidade de portas lógicas, pode conectar-se a um maior número de dispositivos, tornando-se vantajoso para projetos de maior complexidade (CAMPOS, 2014).

De fato, o Arduino Mega 2560, ilustrado na Figura 12, propicia um aumento significativo na funcionalidade de entrada e saída, possuindo 54 pinos digitais, dos quais 14 podem fornecer saída analógica PWM e 16 pinos usados como entradas analógicas. A comunicação é feita com até quatro portas seriais de *hardware*. A comunicação SPI (Serial Peripheral Interface) e o suporte para dispositivos I2C/TWI (Two Wire Interface) estão, também disponíveis. As placas bases podem ser implementadas com placas acessórios, chamadas de *Shields*, as quais podem ser acopladas à placa Arduino (MONK, 2013). Neste projeto, foram utilizados dois *shields*, sendo um a interface de controle para o usuário e o outro para realizar o chaveamento das cargas do módulo.

Figura 12 – Microcontrolador Arduino Mega 2560



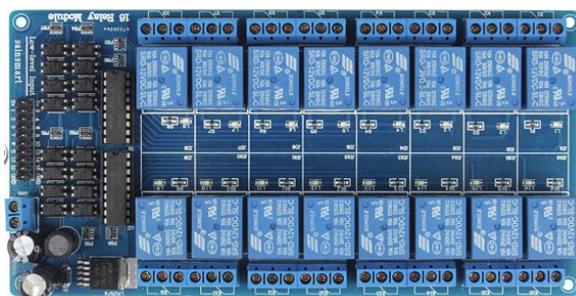
Fonte: [Arduino \(2021\)](#)

4.2.2 Relés de Chaveamento

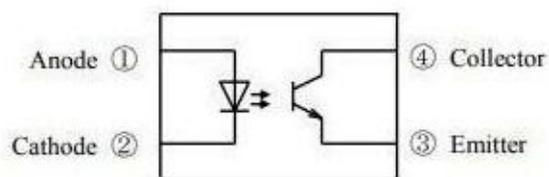
Para realizar a função de chaveamento das cargas, foram utilizados dois módulos de relés, cada um com 16 canais de 5 V, conforme ilustra a Figura 13a. Cada relé suporta corrente alternada de até 10 A e é acionado por meio de saída fotoacoplada.

Figura 13 – Relés de Chaveamento para o Circuito Elétrico

(a) Módulo Relés de 16 Canais



(b) Diagrama Optoacoplador PC817X



Fonte: [Sharp Corporation \(2016\)](#) [Ebay \(2021\)](#)

De modo a evitar completamente o contato elétrico e interferências entre o circuito controlador e o circuito controlado, o módulo de relés contém um circuito integrado optoacoplador para cada relé. Este, por sua vez, é composto por um LED interno ativador na entrada e uma saída foto-acoplada. No caso dos módulos de relés deste projeto, são utilizados optoacopladores PC817x com as saídas foto-transistorizadas, isto é, transistores sensíveis à luz. Um sinal vindo dos pinos lógicos do microcontrolador Arduino é conectado ao ânodo do LED nas portas de entrada do optoacoplador, fazendo com que o mesmo emita

um feixe de luz infravermelha, polarizando a base do fototransistor e forçando a condução entre base e emissor, conforme o esquema da Figura 13b. Dessa forma, eleva-se a tensão BE (base-emissor) a 0,7 V, colocando este semicondutor em saturação. Essa corrente de base para emissor diminui a resistência de canal entre coletor e emissor, produzindo a corrente de coletor que, então, aciona o relé (GONZAGA, 2015).

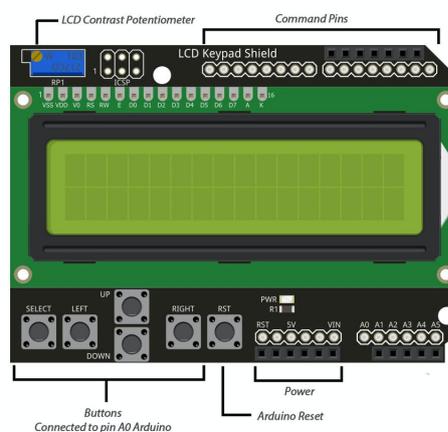
O módulo de 16 canais possui alimentação de 12 V DC e cada relé possui tensão nominal de 5 V, necessitando de uma corrente de 20 mA para ativação de cada um. Dessa maneira, calculou-se para os dois módulos em operação a necessidade de uma fonte de 12 V DC com corrente de 1 A.

4.2.3 Periférico Keypad Shield 16x2

O periférico *Keypad Shield 16x2* foi o responsável por trazer a robustez necessária para este projeto, pois nele pode-se controlar o painel de cargas através dos botões e informações dispostas no menu programável apresentado na tela do mesmo.

Este *shield* foi desenvolvido principalmente para aplicações junto ao Arduino, propiciando um encaixe rápido ao mesmo. Sua principal função é exibir informações junto ao projeto, que serão mostradas no LCD em tempo real. Dentre as funções deste periférico, destaca-se sua utilidade no caso de projetos que trabalham com senhas, ou ainda, com algum tipo de menu que exija navegação, já que conta com botões seletores navegáveis. Este *shield* é constituído por um *display* LCD de 16 caracteres e duas linhas (16x2), possuindo luz de fundo azul com brilho regulável, conforme ilustra a Figura 14 (XAVIER et al., 2021).

Figura 14 – LCD Keypad Shield 16x2



Fonte: Arduino (2021)

Para o projeto do módulo de cargas em questão, o LCD *Keypad Shield* exibe

um menu navegável com os circuitos monofásicos, trifásicos e suas respectivas potências. O usuário pode realizar o controle do mesmo navegando no menu através dos botões dispostos pelo periférico, selecionando as potências no qual deseja operará-lo. A ligação deste componente com a placa Arduino é feita por meio dos pinos digitais 8 ao 13 e ao pino analógico A0, conforme mostra a Tabela 2. No código fonte, o pino A0 é declarado como *digitalRead*, ou seja, é realizada a leitura do mesmo, pois o *shield* envia diferentes sinais de tensão através deste pino quando algum botão do seu teclado é pressionado.

Tabela 2 – Esquema para Conexão do *Keypad Shield* nas Portas Lógicas do Arduino

Portas I/O Arduino	Periférico
8	
9	
10	
11	LCD 16x2 Keypad Shield
12	
13	
A0	

Fonte: Pimentel (2021)

4.2.4 Conexão das Cargas

Conforme visto na seção anterior, os valores das potências exibidas ao operador são uma referência do valor real da potência dissipada. Entretanto, para que seja possível visualizar estes valores, o usuário deve escolher, através do menu principal de navegação exibido no *display*, o circuito no qual o mesmo deseja obter tal informação. Uma vez que o circuito tenha sido selecionado, pode-se acionar, dentro do submenu, os diferentes chaveamentos do circuito elétrico do sistema de potência. A Tabela 3 apresenta as cinco opções de circuito nas quais o operador pode optar e os respectivos valores de referência de potência.

Tabela 3 – Circuitos Monofásicos e Trifásicos e seus Respetivos Valores de Referência de Potência

Carga	Monofásico 220V (W)	Monofásico 110V (W)	Trifásico 220V Y (W)	Trifásico 380V Y (W)	Trifásico 220V Δ (W)
1	60	15	45,0	180	180
2	75	18,75	56,3	225	225
3	100	25	75,0	300	300
4	150	37,5	112,5	450	450
5	250	62,5	187,5	750	750
6	300	75	225,0	900	900
7	360	90	270,0	1080	1080
8	375	93,75	281,3	1125	1125
9	400	100	300,0	1200	1200
10	450	112,5	337,5	1350	1350
11	550	137,5	412,5	1650	1650
12	600	150	450,0	1800	1800
13	675	168,75	506,3	2025	2025
14	700	175	525,0	2100	2100
15	750	187,5	562,5	2250	2250
16	900	225	675,0	2700	2700
17	1000	250	750,0	3000	3000
18	1050	262,5	787,5	3150	3150
19	1200	300	900,0	3600	3600
20	1350	337,5	1012,5	4050	4050
21	1500	375	1125,0	4500	4500
22	1800	450	1350,0	5400	5400

Fonte: Pimentel (2021)

O chaveamento das cargas acontece simultaneamente para as três fases do módulo de cargas. Os relés, S0 até S8, são os responsáveis pelo chaveamento do circuito elétrico, para uma das três fases. Os níveis lógicos dos relés S0 até S8, são mostrados na Tabela 4. Para que as três fases de fato operem realizando o mesmo chaveamento, os relés S9 até S17 possuem os mesmos níveis lógicos apresentados, assim como, S18 até S26.

Tabela 4 – Nível Lógico dos Relés para os Distintos Chaveamentos

Carga	S0	S1	S2	S3	S4	S5	S6	S7	S8
1	1	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0
3	1	0	1	0	0	0	0	0	0
4	1	0	0	1	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	0	0	0	1
7	1	0	0	0	0	0	0	0	1
8	1	1	0	0	0	0	0	0	1
9	1	0	1	0	0	0	0	0	1
10	1	0	0	1	0	0	0	0	1
11	0	0	1	0	0	0	0	0	1
12	1	0	0	0	1	0	0	0	1
13	0	0	0	0	1	0	0	0	1
14	0	0	0	1	1	0	0	0	1
15	0	0	1	0	1	0	0	0	1
16	0	0	0	1	1	0	0	1	1
17	1	0	0	1	1	0	0	1	1
18	1	0	1	1	1	0	1	1	1
19	1	0	1	1	1	0	0	1	1
20	0	0	1	1	1	0	0	1	1
21	0	1	1	1	1	1	0	1	1
22	1	1	1	1	1	1	0	1	1

Fonte: Pimentel (2021)

O circuito monofásico pode ser utilizado conectando-se o equipamento elétrico em qualquer uma das fases do sistema trifásico do módulo de cargas. Assim, operando-se com tensão de alimentação de 110 V, por exemplo, têm-se como referência os valores de potências dissipadas conforme a Tabela 3. Uma vez que o circuito está conectado a uma fonte de tensão, o chaveamento das cargas, executado pelo comando do usuário, inicia com o menor valor de potência. Para que a potência dissipada chegue ao valor máximo, deve-se passar por todos os outros chaveamentos e, portanto, os valores intermediários de potência. Deste modo, os chaveamentos acontecem de forma gradativa, aumentando e diminuindo os valores das potências dissipadas em rampa.

4.2.5 Conexão Carga Trifásica

Tendo em vista os ensaios trifásicos realizados nas máquinas elétricas presentes no LABMAQ-EEL/UFSC, fez-se necessário que o módulo de cargas pudesse operar também em tal condição. Conforme [Alexander e Sadiku \(2013, p. 446\)](#), “Circuitos ou sistemas nos quais as fontes CA operam na mesma frequência, porém, em fases diferentes, são conhecidos como polifásicos”. O sistema trifásico, por sua vez, é um dos casos particulares dos sistemas polifásicos, que por razões técnicas e econômicas, tornou-se padrão em geração, transmissão e distribuição de energia elétrica.

As tensões trifásicas são normalmente produzidas por gerador CA trifásico constituído, basicamente, de um rotor envolto por um estator. A construção mais comum consiste de três bobinas distintas que são dispostas e separadas fisicamente a 120° em torno do estator. À medida que o rotor gira, o seu campo magnético induz tensões nas bobinas, de mesma magnitude, porém defasadas em 120° entre si. Esquematicamente, um sistema trifásico típico é composto por três fontes de tensão, podendo ser visto equivalentemente como três circuitos monofásicos distintos. Deste modo, as fontes de tensão dos três circuitos podem ser conectadas entre si de maneira a gerar uma fonte trifásica em estrela ou triângulo ([ALEXANDER; SADIKU, 2013](#)).

Da mesma forma que nas ligações do gerador, uma carga trifásica também pode ser conectada em estrela ou triângulo, conforme ilustra a Figura 15. Considera-se que uma carga conectada em estrela ou triângulo está desequilibrada se as impedâncias por fase não forem iguais em magnitude ou fase ([ALEXANDER; SADIKU, 2013](#)). Para o módulo de cargas em questão, utilizou-se lâmpadas idênticas em todas as fases visando, principalmente, que fosse possível trabalhar com a carga equilibrada em todos os níveis de potência.

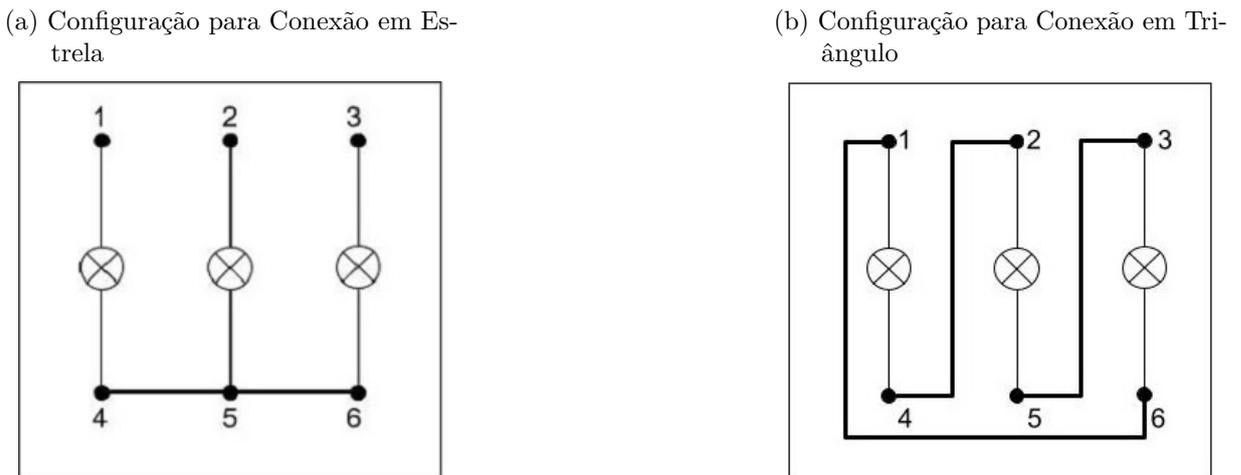
Figura 15 – Configurações Possíveis para Cargas Trifásicas



Fonte: Alexander e Sadiku (2013)

Cada uma das três fases possui um borne de entrada e saída na parte superior do módulo e, para os ensaios realizados com alimentação trifásica, o usuário deve realizar a conexão das cargas de acordo com a atividade proposta. Para operar em estrela, um dos bornes de cada carga devem estar conectados entre si, formando um ponto comum entre os mesmos, enquanto que os outros três bornes estarão conectados às três fases da fonte de alimentação trifásica, respectivamente, conforme a ilustra a Figura 16a. Quando se deseja trabalhar com as cargas conectadas em triângulo, um dos bornes das cargas deve estar conectado a um borne de outra carga e as três fases do sistema trifásico são conectadas aos nós de junção das cargas, de acordo com a Figura 16b.

Figura 16 – Bornes de Conexão para o Painel de Cargas



Fonte: Arquivo Pessoal

4.2.5.1 Trifásico Estrela 220 V e 380 V

Uma vez que se tem a conexão em estrela realizada corretamente nos bornes do módulo de cargas, o usuário pode, da mesma forma que no monofásico, selecionar por

meio do *display*, os valores de potência que deseja operar. Neste caso, pode-se trabalhar com dois valores distintos na tensão de alimentação, sendo estes 220 V e 380 V. Para a conexão das cargas em estrela, sabe-se que a corrente de linha e de fase possuem a mesma magnitude e, portanto, para ambos níveis de tensão, as cargas estarão operando dentro dos valores nominais. Os valores de referência de potência apresentados no submenu do *display* podem ser observados na Tabela 3.

4.2.5.2 Trifásica Triângulo 220 V

O acionamento das potências com as cargas trifásicas conectadas em triângulo é realizado da mesma maneira que em estrela, porém com somente um nível de tensão de alimentação. Para uma carga conectada em triângulo, as correntes de linha possuem magnitude raiz de três vezes maior que às de fase e, portanto, para tensões maiores que 220 V, as cargas iriam operar acima dos valores nominais. A Tabela 3 apresenta os valores de referência de potência para tal circuito.

4.2.6 Programação do Código de Controle

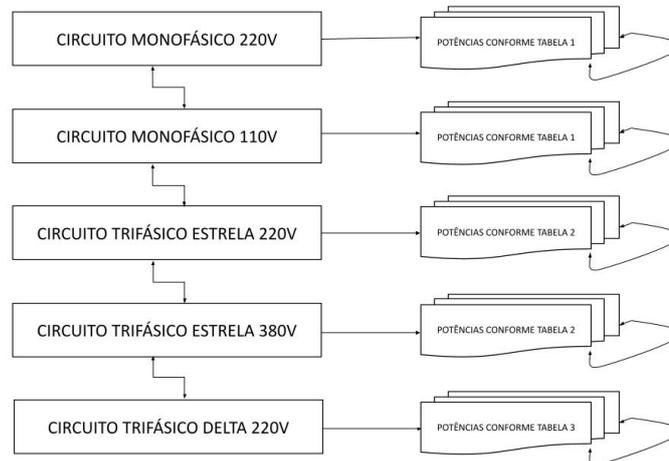
Para implementar programas no ambiente Arduino, utiliza-se um código fonte próprio de referência que tem como base a linguagem de programação C++. Este mantém a sintaxe clássica da linguagem base, como por exemplo, na declaração de variáveis, tipos de operadores, estrutura de dados vetores, definição e utilização de ponteiros, estruturas de controle e em muitas outras características da linguagem C++. (GUEDES, 2018).

As operações realizadas pelo Arduino se resumem, principalmente, em sinais elétricos. A lógica por trás dos impulsos e sinais transmitidos e interpretados pela plataforma dependem de programas inseridos em seu microcontrolador. Os comandos e programas são escritos no computador através do ambiente de desenvolvimento integrado, em inglês *Integrated Development Environment* (IDE), o qual permite criar, alterar e converter códigos em instruções que o Arduino consiga compreender e executar. Em resumo, tudo o que será realizado pelo Arduino, seja através do microcontrolador ou de periféricos a ele conectados, deve ser programado em códigos compilados e inseridos ao microcontrolador por intermédio da IDE (FERRONI et al., 2015).

Para o projeto em questão, foi elaborada uma interface de controle para o usuário que consiste de um menu de acionamento para as cargas do módulo. O diagrama de blocos da Figura 17 ilustra as opções no menu principal de navegação exibido no *display* LCD do periférico *keypad shield*. Em um primeiro momento, o usuário pode escolher, navegando através dos botões *up* e *down* do *keypad shield*, entre os circuitos monofásico 110 V e 220 V, trifásico estrela 220 V e 380 V ou trifásico delta 220 V. Uma vez selecionado o circuito, usando o botão *select*, tem-se o submenu com os 22 valores de potência de acordo com as tabelas apresentadas anteriormente, para cada circuito específico. Cada potência é

mostrada no *display* individualmente e o submenu é controlado por meio dos botões *right* e *left*.

Figura 17 – Interface de Controle para o Usuário



Fonte: Arquivo Pessoal

O código do programa divide-se em três funções principais rodando em *loop* e chamando novas sub funções, quando necessário. As funções *KeyboardRead*, *DispMenu* e *ChangeMenu* são as responsáveis, respectivamente, por realizar a leitura dos botões do *keypad shield*, exibir os menus de navegação no LCD, realizando o devido chaveamento das cargas e realizar a troca dos menus ajustando os valores das *flags* dos botões. O trecho do Algoritmo 1 mostra o *loop* principal do programa. Desta forma, para uma compreensão mais clara, estas funções serão apresentadas separadamente, nas próximas seções.

Algoritmo 1 – Funções do *Loop* Principal no Código Fonte

```
void loop()
{
  //-- Funções do loop principal--
  keyboardread();
  changeMenu();
  dispMenu();
} //Fim do loop
```

Fonte: Pimentel (2021)

4.2.6.1 Lógica de Controle da Função *KeyboardRead*

Para a lógica de controle dos botões, optou-se por criar uma variável booleana específica para os botões *right*, *up*, *down*, *left* e *select*. As variáveis foram declaradas, respectivamente, como: *butt01*, *butt02*, *butt03*, *butt04* e *butt05*. Cada variável pode conter apenas um de dois valores. No código de controle do módulo de cargas, as variáveis podem conter os valores hexadecimais 0x00 ou 0x01.

Quando um botão do *keypad shield* é pressionado, um sinal de tensão específico é enviado para a porta analógica A0 do Arduino, que possui um conversor AD de 10 *bits*. Declarou-se, então, a função *keyboardRead*, responsável por verificar se algum botão foi pressionado pela verificação do nível de tensão do sinal na porta analógica. Os valores dos sinais de tensão de cada botão foram retirados do *datasheet* do periférico *keypad shield* e convertidos para o respectivo valor digital de 10 bits, conforme a Tabela 5. Se algum botão foi pressionado, a função *keyboardRead* verifica qual é o valor digital na porta analógica e atribui à variável booleana o valor hexadecimal 0x01 do respectivo botão. Em seguida, testa-se se o botão pressionado foi solto observando-se possível alteração do valor digital na porta analógica. Se houve alteração, o botão foi solto e a variável é definida com valor hexadecimal 0x00. O trecho do Algoritmo 2 mostra a lógica citada para o acionamento do botão *right*.

Tabela 5 – Conversão AD dos Botões *Keypad Shield*

Botão	Tensão Analógica (V)	Saída Digital 10 bits
Right	0	0
Up	0,71	145
Down	1,61	329
Left	2,47	505
Select	3,62	741

Fonte: Pimentel (2021)

Dessa forma, realizou-se a leitura do teclado do periférico *keypad shield* para todos os botões. Ressalta-se que a função *keyboardRead* necessitou ser chamada em todo trecho de código fora do *loop* principal.

Algoritmo 2 – Lógica de Controle dos Botões Keypad Shield

```
// --- Variáveis Globais ---
boolean right = 0x00, butt01 = 0x00,
up = 0x00, butt02 = 0x00,
down = 0x00, butt03 = 0x00,
left = 0x00, butt04 = 0x00,
select = 0x00, butt05 = 0x00;

void keyboardRead()
{
    //Variável para leitura recebe valor AD de A0
    adc_value = analogRead(A0);

    // --- Testa se os botões foram pressionados ---
    // Se foi pressionado, seta a respectiva flag
    if (adc_value < 50) butt01 = 0x01;
    else if (adc_value > 103 && adc_value < 200) butt02 = 0x01;
    else if (adc_value > 250 && adc_value < 380) butt03 = 0x01;
    else if (adc_value > 450 && adc_value < 550) butt04 = 0x01;
    else if (adc_value > 650 && adc_value < 800) butt05 = 0x01;

    // --- Testa se os botões foram liberados ---
    if (adc_value > 50 && butt01) //Botão right solto e flag butt01 setada?
    {
        //Se sim...
        butt01 = 0x00; //Limpa flag butt01
        right = 0x01; //Seta flag right
    }
}
```

Fonte: Pimentel (2021)

4.2.6.2 Lógica de Controle da Função *dispMenu* e *changeMenu*

Para o estudo da lógica das funções *dispMenu* e *changeMenu*, devido ao grande número de linhas de código, será considerado que o usuário optou por operar com o módulo de cargas com o circuito monofásico de 220 V na potência de 300 W, por exemplo. Assim, faz-se possível discorrer sobre a lógica do programa seguindo o caminho de um objetivo de acionamento de circuito e carga específicos. O Algoritmo 3 ajuda a entender o exemplo.

Algoritmo 3 – Lógica de Controle para a Função *dispMenu*

```
void dispMenu()
{
  //--Inicia comando switch(case)--
  switch(menu)
  {
    case 0x01:
      disp.setCursor(15,0); //Posiciona cursor 15x0 do LCD
      disp.write((byte)0x01); //Escreve o caract er de seta p/ baixo

      disp.setCursor(0,0); //Posiciona cursor 0x0 do LCD
      disp.write((byte)0x02); //Escreve caract er de seta p/ lado '>'

      disp.setCursor(1,0); //Posiciona cursor 1,0 do LCD
      disp.print(F("Monof. 110V  ")); //Escreve 'Monof. 110V' no LCD
      disp.setCursor(0,1); //Posiciona cursor 0,1 do LCD
      disp.print(F("Press. Select  ")); // Escreve 'Press. select' no LCD

      if (select == 0x01){ //Verifica se select foi pressionado
                          //Se sim...
        dispsubmenu_1(); //Chama a fun  o dispsubmenu_1
      }

      break;
  }
}
```

Fonte: Pimentel (2021)

Ap s a inicializa  o do microcontrolador, inicia-se a fun  o *loop* principal. As fun  es *keyboardRead* e *changeMenu* dependem ambas de a  es realizadas nos bot es do *keypad shield*, enquanto a fun  o *dispMenu* exibe imediatamente a primeira op  o de escolha dos circuitos da lista. Esse fluxo   controlado por meio do comando *switch (case)*, o qual executa o comando do *case 0x01* e, se nenhuma decis o for tomada, ao encontrar o comando *break*, volta para o *loop* principal. Dentro do primeiro *case*, exibe-se uma seta para baixo, logo ap s   mensagem “Monof. 110V”, indicando para o usu rio que o bot o *down* pode ser pressionado para que uma nova op  o de circuito seja exibida. Logo abaixo da mensagem, na segunda linha do display LCD, encontra-se a mensagem “Press. Select”. Intuitivamente, pressionado o bot o *select*, o usu rio escolhe operar no circuito em quest o. Visto que essa decis o tenha sido tomada, o *keyboarRead* define a *flag* de *select* em 0x01 e a fun  o *dispsubmenu_1*   chamada. Por outro lado, se o bot o *down* for pressionado, a fun  o *keyboardRead* do *loop* principal define a *flag* do bot o em quest o em 0x01 e, ent o, a fun  o *changeMenu* entra em a  o.

A função *changeMenu* verifica se as *flags* dos botões *up* ou *down* foram definidas. Se a *flag down* for definida, o *case menu*, do comando *switch(menu)*, é incrementado e a *flag down* atribuída com 0x00. Alternativamente, se *up* for pressionado, o *case menu* é decrementado, conforme mostra o trecho do Algoritmo 4. Dessa forma, cada vez que se pressiona *down* ou *up*, troca-se o menu para o *case* posterior ou anterior, respectivamente. Na sequência é apresentado o Algoritmo 5, que mostra a lógica de controle para a função *changeMenu*.

Algoritmo 4 – Lógica de Controle para a Função *changeMenu*

```
void changeMenu()
{
    if(up == 0x01)        //Se botão up pressionado
    {
        up = 0x00;        //Seta flag de up em 0x00
        menu--;           //Decrementa o menu da função switch(menu)
        if(menu < 0x01) menu = 0x01; //Primeira opção do menu
    }
    if(down == 0x01)     //Se o botão down pressionado
    {
        down = 0x00;      //Seta flag de down em 0x00
        menu++;           //Incrementa o menu da função switch(menu)
        if(menu > 0x05) menu = 0x05; //Última opção do menu
    }
} //Fim da função changeMenu
```

Fonte: Pimentel (2021)

Retornando ao objetivo inicial de acionamento da carga de 300 W do circuito monofásico de 220 V, pressiona-se o botão *down* uma vez e *select*, posteriormente. Consequentemente, conforme visto, a função, neste caso *dispsubmenu_2*, é chamada e o usuário acessa o submenu para o acionamento das potências do circuito monofásico de 220 V do módulo de cargas. Inicialmente, a função *dispsubmenu_2*, conforme mostra o trecho do Algoritmo 5, fixa o valor de *select* em 0 e declara uma nova variável $x = 1$. Posteriormente, utiliza-se o comando *while* com a expressão $x = 1$, como condição para sua execução. Enquanto a condição for verdadeira, o comando *while* segue executando duas tarefas principais dentro do seu laço. A primeira ação utiliza a mesma lógica da função *changeMenu*, com o comando *switch(case)* para impressão das mensagens no display LCD ao usuário e, também, para chamar a função que realiza os chaveamentos dos relés no circuito de cargas do módulo. Paralelamente, a cada ciclo de execução do *switch(case)*, verifica-se, por meio da função *keyboardRead*, se houve acionamento das teclas *right* ou *left*. No primeiro *case* (0x01) imprime-se a mensagem “Press. Select p/ Voltar ou >” e não

se ativa nenhuma carga do circuito elétrico. Dessa forma, se o usuário pressionar *select*, altera-se o valor da variável *x* para 0 e se finaliza o laço *while* retornando, assim, para o *loop* principal. Ressalta-se que, ao voltar para o *loop* principal, devido ao valor de menu do *switch(case)* principal ainda estar com o valor incrementado, retorna-se imediatamente para o *case* 0x02, realizando o mesmo caminho de volta.

Algoritmo 5 – Lógica de Controle para a Função *changeMenu*

```
void dispsubmenu_2()
{
    select = 0x00;
    x=1;

    while (x == 1)
    {
        select = 0x00;
        if(right == 0x01)
        {
            right = 0x00;
            submenu_2 ++;
            if(submenu_2 > 0x17) submenu_2 = 0x17;
        }
        if(left == 0x01)
        {
            left = 0x00;
            submenu_2 --;
            if(submenu_2 < 0x01) submenu_2 = 0x01;
        }

        switch(submenu_2)
        {
            case 0x01:
                keyboardRead();

                if (select == 0x01){
                    x=0;
                    up = 0x00;
                    down = 0x00;
                }
                disp.setCursor(0,0);
                disp.print(F("Press. Select p/"));
                disp.setCursor(0,1);
                disp.print(F("Voltar   ou   >"));
                semcarga();
                break;
        }
    }
}
```

Fonte: Pimentel (2021)

Em contrapartida, ainda dentro da lógica da função *dispsubmenu_2*, quando o botão *right* é pressionado, incrementa-se o valor do *case* do *switch(submenu_2)* e se executa os comandos, conforme mostra o trecho do Algoritmo 6. A partir deste *case*, na primeira linha

do *display*, a mensagem “Monofásico 220 V” é impressa enquanto que na linha inferior, é exibido o valor da potência que o chaveamento está executando. Para o envio dos sinais aos pinos do microcontrolador, é chamada a função *carga_1* que, usando o do comando *digitalWrite*, determina o nível lógico que é enviado aos relés. Nesse estágio, pressionando o botão *right* duas vezes, entra-se no *case* 0x04, que executa a função *carga_3*, exibindo o valor de potência de 300 W no display e chaveando o circuito de cargas do módulo de acordo com tal valor de potência. Sendo assim, conclui-se o objetivo de ativação da carga de 300 W do circuito monofásico de 220 V, proposto no início desta seção.

Algoritmo 6 – Lógica de Controle para a Função *changeMenu*

```
case 0x02:

    keyboardRead();

    disp.setCursor(0,0);
    disp.write((byte)0x02);

    disp.setCursor(1,0);
    disp.print(F("Monofasico 220V"));
    disp.setCursor(0,1);
    disp.print(F("< P=60W          >"));

    carga_1();

    break;
```

Fonte: Pimentel (2021)

É importante ressaltar que, dentro da lógica de cada função *dispsubmenu*, programou-se para que os chaveamentos de potência somente sejam realizados em rampa, tanto para aumentar, quanto diminuir a carga. Isso significa que, para chegar no valor de maior potência, faz-se necessário passar por todas as potências intermediárias. Uma vez operando no último valor de potência, para que seja possível sair do submenu ou desligar a carga, o usuário deve voltar por todas potências até chegar, novamente, ao *case* 0x01. A mesma lógica se repete para a ativação das cargas de todos os outros circuitos elétricos.

4.2.6.3 Lógica de Controle para o Modo Avaliação

Apesar de não ter sido citado nos capítulos anteriores, devido a aplicação didática do módulo de cargas no LABMAQ-EEL/UFSC, foi realizada uma programação específica para operar no modo avaliação. Quando o mesmo estiver neste modo, o usuário visualiza

apenas a mensagem “Carga 1”, por exemplo. Dessa forma, o discente realiza os ensaios e medições sem saber o valor de referência de potência.

O modo “avaliação” é ativado com auxílio de uma chave conectada a porta lógica 26 do microcontrolador. Se a mesma estiver acionada, um sinal de 5 V é lido quando o microcontrolador for iniciado, exibindo a mensagem “Modo Avaliacao” no display e, posteriormente, chamando a função *modoavaliacao*, conforme mostra o trecho do Algoritmo 7. A mesma lógica do *switch(case)*, utilizada para o menu principal, é implementada para o modo “avaliação”, mas com a diferença de que não há submenu neste caso. O usuário tem apenas a opção de pressionar os botões *right* e *left* para navegar entre os diferentes valores de carga. A carga a ser operada ficará a cargo do docente, assim como a atividade a ser realizada.

Algoritmo 7 – Lógica de Controle para o Modo Avaliação

```
disp.setCursor(0,0);
disp.print (F(" Bem Vindo  "));
disp.setCursor(0,1);
disp.print(F("Painel de Carga"));

delay(200);
disp.clear();

if (digitalRead(26) == HIGH){

    disp.setCursor(0,0);
    disp.print (F(" Modo Avaliacao  "));
    delay(200);
    disp.clear();

    modoavaliacao();
}
while (select == 0x00){

keyboardRead();

disp.setCursor(0,0);
disp.print(F("Pressione Select"));
disp.setCursor(0,1);
disp.print(F(" Para Iniciar  "));
}
}
```

Fonte: Pimentel (2021)

5 SIMULAÇÃO COMPUTACIONAL DO SISTEMA DE COMANDO

Utilizar simulações computacionais para análise das condições de funcionamento de circuitos eletrônicos é de extrema importância quando se pretende projetar circuitos microcontrolados. Ao realizar a simulação, torna-se possível verificar, dentre várias configurações de circuitos disponíveis, qual terá desempenho mais adequado à sua aplicação. Assim, não há necessidade de construir diversos protótipos e fazer testes e medições laboratoriais, tornando desnecessário adquirir ou obter componentes para montagens experimentais (MEHL, 2010). Para o caso do circuito microcontrolado do módulo de cargas deste projeto, tornou-se possível, através da simulação, testar diferentes lógicas de programação, com objetivo de otimizar a navegação do menu de controle exibido no *display*.

5.1 SIMULAÇÃO MÓDULO DE CARGAS

A escolha do *software* para simulação levou em conta, principalmente, a disponibilidade de material didático aplicável aos objetivos do projeto. Dentre os diversos simuladores computacionais disponíveis, optou-se, então, por utilizar o *software* PROTEUS.

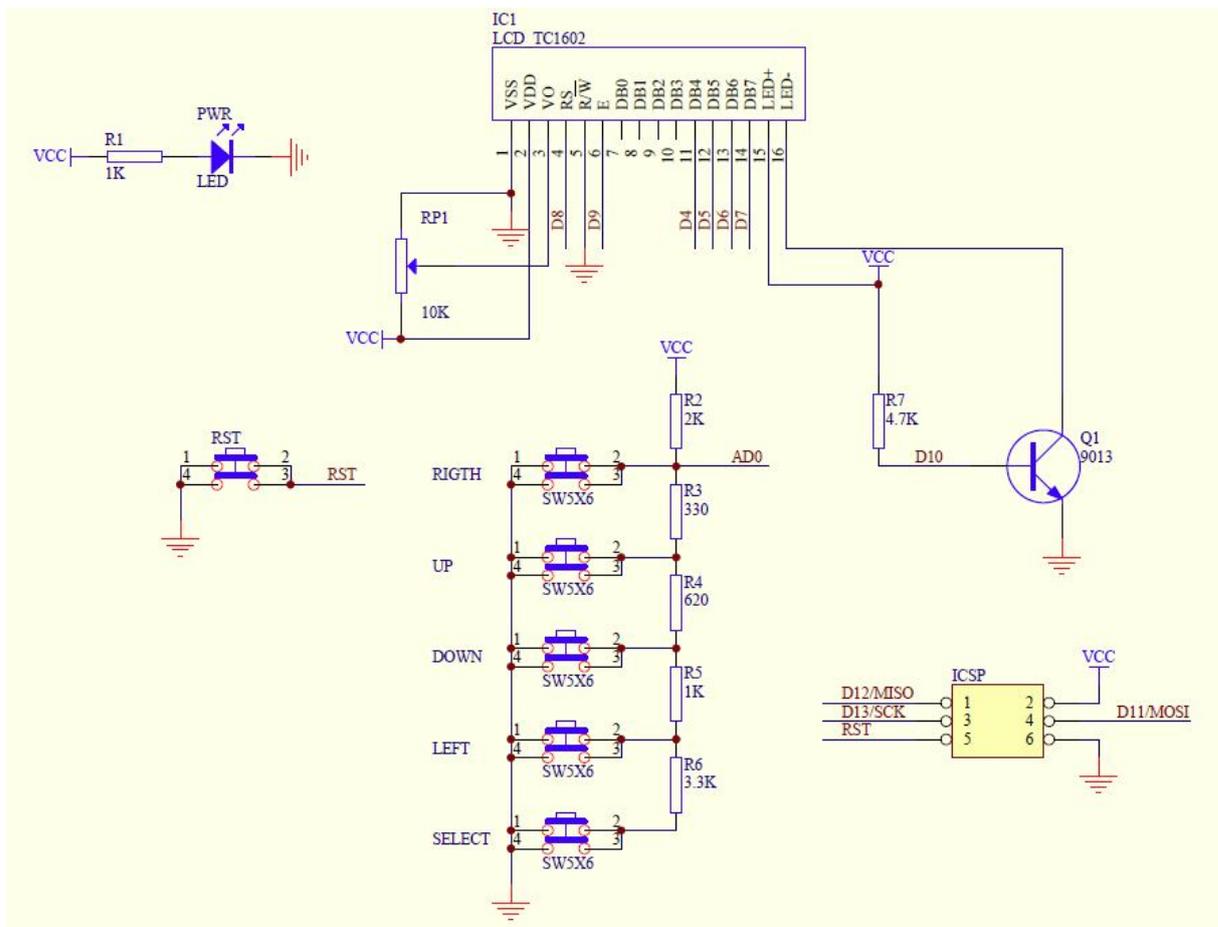
Dentro do *software* PROTEUS, encontra-se o ISIS - *Intelligent Schematic Input System* (Sistema de Entrada Esquemático Inteligente), ferramenta para desenvolvimento de esquemáticos e o ARES - *Advanced Routing and Editing Software* (Roteamento Avançado e Edição de *Software*), ferramenta que permite o desenvolvimento de *layouts* de circuitos impressos PCB (*Printed Circuit Board*) (SANTOS, 2014). Para este projeto, utilizou-se apenas a ferramenta ISIS, pelo fato da mesma apresentar uma biblioteca vasta de componentes eletrônicos, incluindo o microcontrolador utilizado na placa Arduino. Uma vez que o código escrito no *software* IDE, citado anteriormente, foi compilado sem erros, torna-se possível gerar um arquivo .hex (hexadecimal) a partir do mesmo e integrá-lo ao microcontrolador simulado no ISIS. Ainda no mesmo ambiente, fez-se possível desenhar o circuito elétrico das cargas do módulo, com o chaveamento sendo realizado com auxílio da lógica de controle do microcontrolador. Dessa forma, realizou-se a simulação da navegação do menu exibido no *display* LCD 16x2 e verificou-se os valores de potências dissipadas com o uso das ferramentas de medição do *software*.

5.1.1 Simulação Menu Display Keypad Shield

Para o *display* LCD 16x2, utilizou-se o componente LM016L da biblioteca ISIS e, para a conexão dos botões *right*, *left*, *up*, *down* e *select*, seguiu-se o diagrama elétrico presente no *datasheet* do *Keypad Shield*, conforme ilustra a Figura 18. Internamente, o componente *keypad Shield* possui um circuito divisor de tensão, que consiste, basicamente,

de resistores associados em série, com os botões posicionados de forma a realizar a conexão com o terra, quando pressionado. Assim, aplica-se uma tensão de entrada DC de 5 V e, na saída, obtém-se uma fração fixa da tensão de entrada. Para o componente em questão, são utilizados cinco resistores no circuito interno de divisor de tensão, resultando em cinco níveis de tensão distintos na saída.

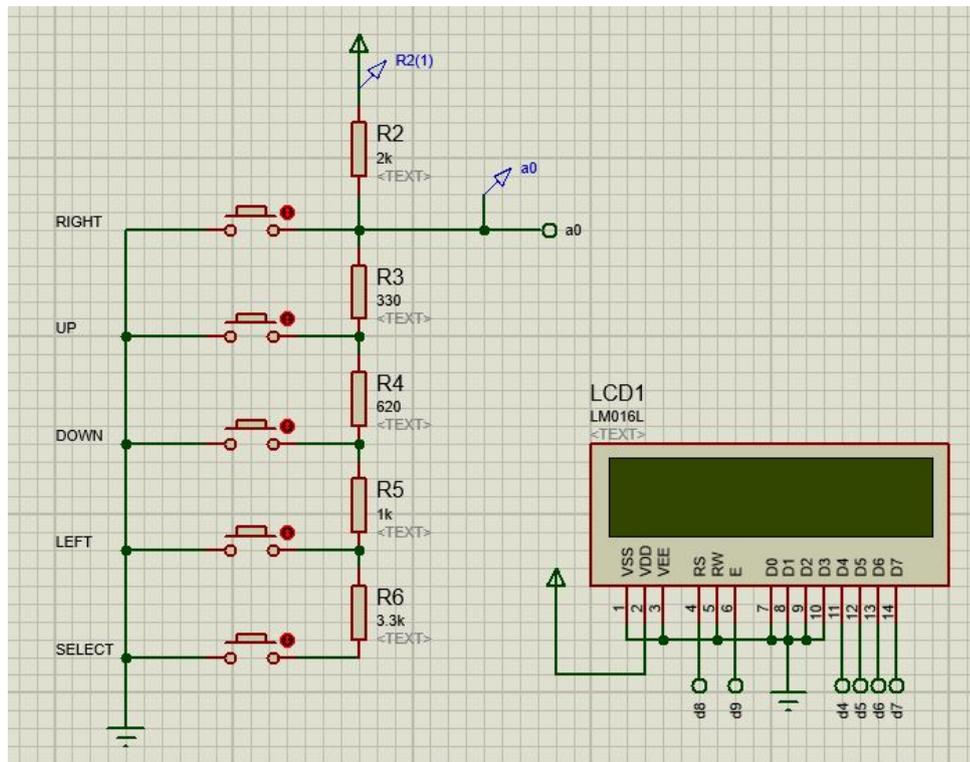
Figura 18 – Diagrama Elétrico do *datasheet* do LCD Keypad Shield



Fonte: D-Robotics (2011)

Desenhou-se, assim, o circuito elétrico no ambiente ISIS com os mesmos valores dos resistores do diagrama elétrico do *datasheet*, obtendo o circuito da Figura 19. A conexão do periférico com o microcontrolador Arduino seguiu o esquema de ligação da Tabela 2, que utilizou as instruções de conexão presentes no *datasheet* do componente em questão. A ferramenta *Terminals Mode*, que substitui a conexão com fio ponto a ponto, foi utilizada no ambiente de simulação, assegurando mais organização e clareza no desenho do projeto.

Figura 19 – Circuito Elétrico LCD *Keypad Shield* Projetado no Ambiente ISIS



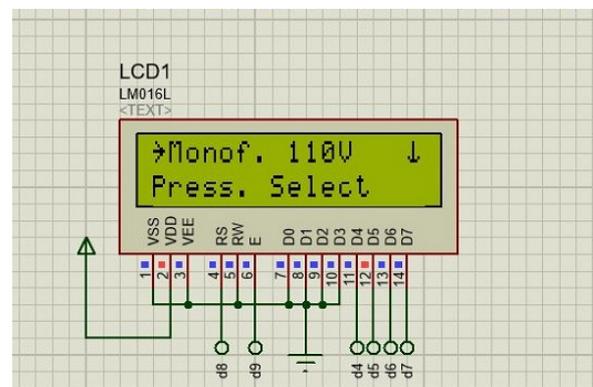
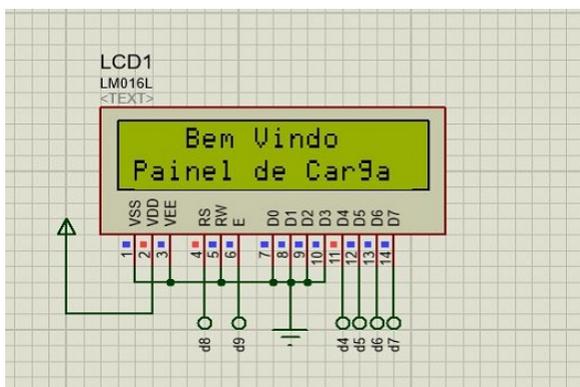
Fonte: Arquivo Pessoal

Uma vez que se desenhou o circuito com as devidas conexões, foi possível simular o *display* LCD e navegar no menu do projeto do módulo de cargas, conforme ilustra a Figura 28. Dessa forma, simulando o *display* LCD, tornou-se possível otimizar, por meio do *software* IDE, a lógica de programação para que a navegação no menu e submenu do projeto se tornasse simples e intuitiva.

Figura 20 – Simulação Display LCD 16x2 *Keypad Shield*

(a) Inicialização do Painel de Cargas

(b) Menu de Navegação Principal

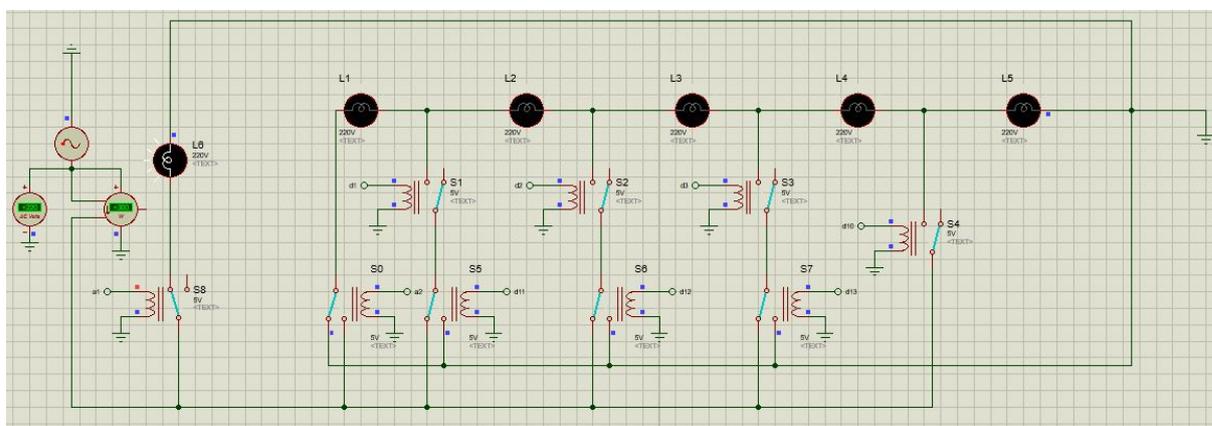


Fonte: Arquivo Pessoal

5.1.2 Simulação Circuito Elétrico Monofásico

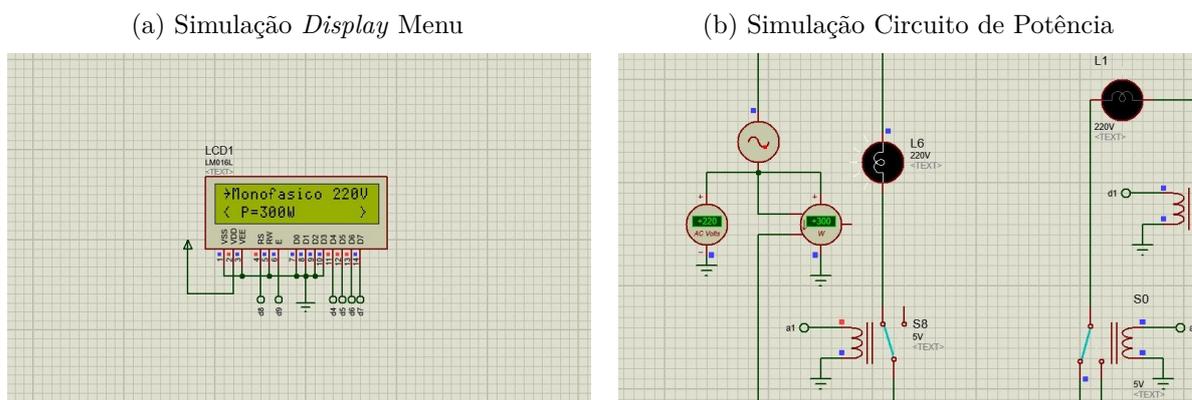
Para simulação do circuito elétrico, utilizou-se os componentes *lamp*, *relay* e *alternator* para simular, respectivamente, as cargas, relés de chaveamento e a fonte de tensão. O circuito elétrico da Figura 11, apresentado no capítulo anterior, foi desenhado no ambiente ISIS, com as chaves (S0, S1, S2,...) sendo substituídas pelo componente *relay* da biblioteca PROTEUS, conforme ilustra a Figura 21. Para as cargas, utilizou-se a resistência $R=161,33 \Omega$, que representa as duas lâmpadas incandescentes de 150 W associadas em série. As alimentações dos relés foram conectadas às portas lógicas do microcontrolador, tornando possível o controle de ativação das cargas através do *keypad shield* simulado.

Figura 21 – Circuito de Potência Completo



Fonte: Arquivo Pessoal

Utilizou-se a ferramenta *wattmeter* (wattímetro) no PROTEUS para verificar se a potência dissipada no circuito simulado corresponderia àquela exibida no menu do display LCD. Na Figura 22a, observa-se o menu exibindo a informação monofásico 220 V/300 W enquanto a Figura 22b mostra a medição no wattímetro para o ponto de operação em questão. Dessa forma, certificou-se o correto funcionamento da lógica de programação e do circuito elétrico de potência para todas as outras cargas do circuito monofásico 110 V e 220 V.

Figura 22 – Simulação *Display* e Carga Operando em Circuito Monofásico 220 V/300 W

Fonte: Arquivo Pessoal

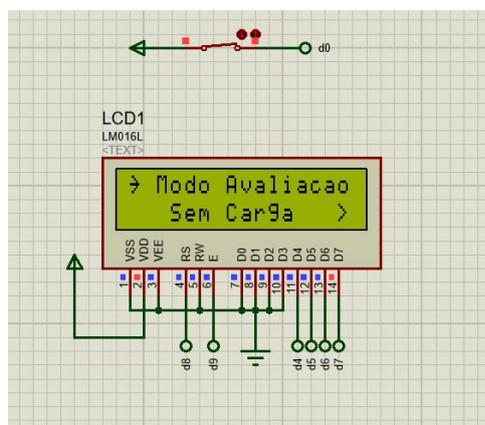
5.1.3 Simulação Circuito Elétrico Trifásico

Pelo fato de se tratar de um sistema trifásico equilibrado o mesmo pode ser visto como 3 circuitos monofásicos operando juntamente, conforme estudado nos capítulos anteriores. Uma vez que a conexão estrela ou triângulo entre os circuitos de carga foram realizadas corretamente pelo operador e a simulação funcione sem erros para o circuito monofásico garante-se também o correto funcionamento para operação do sistema trifásico.

5.1.4 Simulação Modo Avaliação

Tendo as cargas já sido testadas nas simulações dos circuitos monofásicos, tornou-se necessário para o modo “avaliação” somente a verificação do menu de navegação. Para realizar a simulação do mesmo, utilizou-se uma chave com tensão de alimentação de 5 V DC na entrada e a saída conectada ao microcontrolador. Conforme visto no capítulo anterior, quando a chave estiver fechada, o sinal é lido na porta do ARDUINO que, ao iniciar, exibe o menu de modo avaliação no *display* LCD. A mudança das cargas segue a mesma lógica dos menus já vistos anteriormente, mas com a diferença de que não está disponível a informação dos valores das potências de referência para cada configuração das cargas, conforme ilustra a Figura 23. Sendo assim, verificou-se o correto funcionamento de todos os menus e seus respectivos chaveamentos de carga no circuito de potência do módulo, com auxílio das simulações realizadas no ambiente ISIS PROTEU.

Figura 23 – Simulação Modo Avaliação



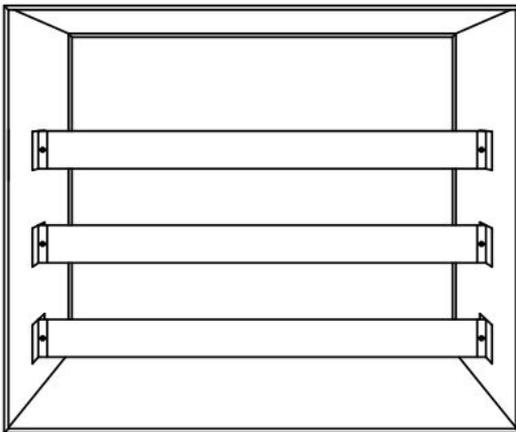
Fonte: Arquivo Pessoal

6 IMPLEMENTAÇÃO DO PROJETO

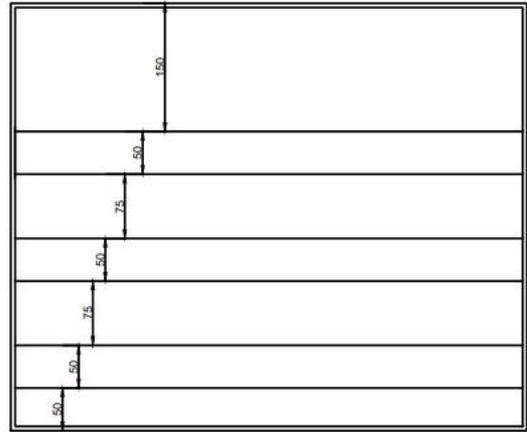
A implementação do projeto foi realizada no LABMAQ-EEL/UFSC pelo técnico de laboratório Cristian Franzoi Mazzola. Primeiramente, realizou-se o projeto do *layout* externo e interno do módulo de cargas, com dimensões para conter as 36 lâmpadas mais o circuito microcontrolador. O resultado foi uma caixa retangular de aço com altura, comprimento e largura de, respectivamente, 600 x 400 x 500 mm, mostrado na Figura 24 (*layout* interno) e na Figura 25 (*layout* externo).

Figura 24 – Layout Interno do Painel de Cargas

(a) Vista Lateral em Perspectiva sem as Portas



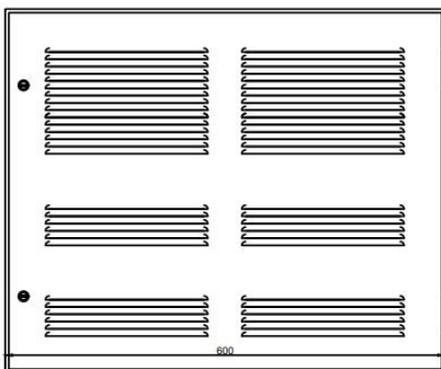
(b) Vista Lateral



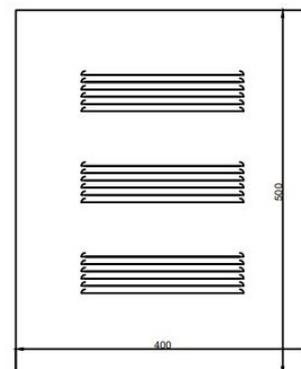
Fonte: Mazzola (2019)

Figura 25 – Layout Externo Painel de Cargas

(a) Portas Direita e Esquerda



(b) Vista Frontal e Traseira



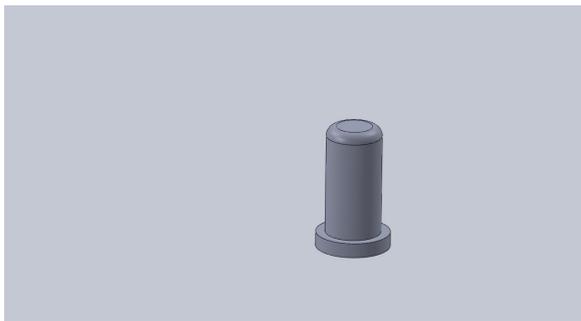
Fonte: Mazzola (2019)

O painel contou com três barras horizontais internas, separadas de 50 mm, conforme ilustra as Figuras 24a e 24b. Em cada barra, foram conectadas doze lâmpadas, sendo seis de cada lado, totalizando, nas três barras, 36 lâmpadas. Cada lâmpada foi ligada com os fios de alimentação fase e neutro, com seção (bitola) $1,0 \text{ mm}^2$, aos respectivos módulos de relés. As tampas laterais foram projetadas com aletas ao longo de sua superfície, objetivando aumentar a troca térmica com o ambiente por convecção, conforme mostra a Figura 26.

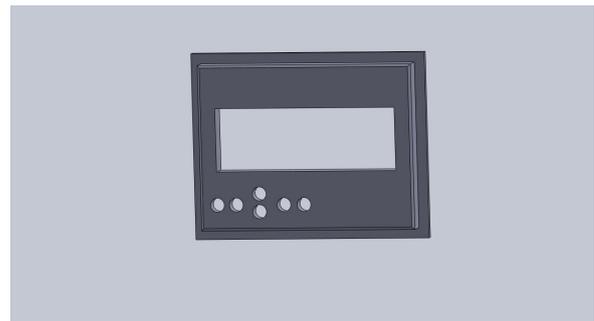
Na tampa superior, foi realizado um corte com as dimensões da carcaça de encaixe do *display*, conforme ilustrada a Figura 26b. O *display* foi posicionado na extremidade da placa superior do módulo de cargas. Logo acima, foram colocados os bornes R, R', S, S' e T, T' de entrada e saída de cada respectivo circuito monofásico. Os botões do teclado *keypad shield* foram projetados em escala adequada para a carcaça de encaixe do *display* e impressos em ABS (tipo de plástico rígido) em impressora 3D, seguindo *layout* da Figura 26a.

Figura 26 – Layout da Peça para o Encaixe do *Keypad Shield*

(a) Botões Impressos em ABS com Impressora 3D

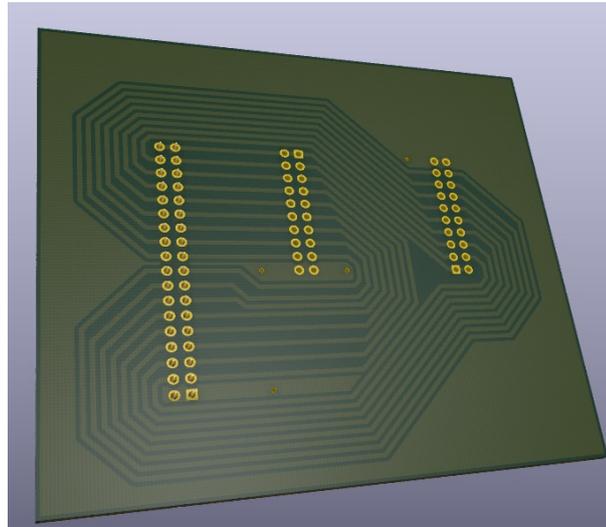


(b) Carcaça de Encaixe do Display



Fonte: [Cardenuto \(2019\)](#)

Com intuito de organizar o cabeamento e melhorar a conexão das portas lógicas do Arduino com as placas de relés, projetou-se um circuito impresso dividindo as 32 portas lógicas do Arduino em dois pontos de conexão para cabo *flat* de 16 fios, conforme ilustra a Figura 27. As conexões dos relés nas portas lógicas da placa do microcontrolador foram realizadas de acordo com a Tabela 6.

Figura 27 – Placa *Split* Para Conexão Arduino-Relés

Fonte: Duarte (2019)

Tabela 6 – Esquema para Conexão dos Módulos de Relés nas Portas Lógicas do Arduino

Portas I/O Arduino	Chaves Circuito Elétrico	Periférico
27	S0	
28	S1	
29	S2	
30	S3	
31	S4	
32	S5	
33	S6	Módulo Relés 1
34	S7	
35	S8	
36	S9	
37	S10	
38	S11	
39	S12	
40	S13	
41	S14	
42	S15	
43	S16	
44	S17	
45	S18	
46	S19	Módulo Relés 2
47	S20	
48	S21	
49	S23	
50	S24	
51	S25	
52	S26	
53	S27	

Fonte: Pimentel (2021)

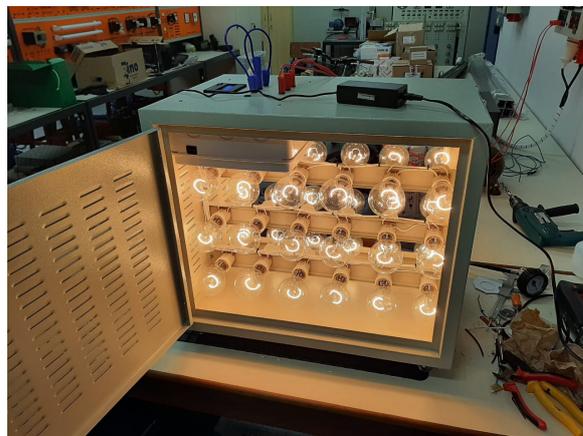
Após realizadas as conexões dos periféricos ao microcontrolador e às cargas nos módulos de relés, seguindo os esquemas de conexão, obteve-se o módulo de cargas conforme ilustram as fotos da Figura 28. Ressalta-se que, na Figura 28b, o módulo está operando com o sistema trifásico, certificando, assim, o funcionamento dos três circuitos monofásicos.

Figura 28 – Implementação Módulo de Cargas

(a) Montagem Módulo de Cargas



(b) Módulo de Cargas em Operação



Fonte: Arquivo Pessoal

7 POSSÍVEIS AJUSTES E SUGESTÕES PARA FUTUROS TRABALHOS

Sabe-se, dos capítulos anteriores, que as potências exibidas no *display* LCD são somente uma referência da potência real dissipada pela carga devido às características eletrofísicas do filamento interno da lâmpada incandescente, quando operando em diferentes temperaturas. Se o operador do módulo de cargas deseja realizar os ajustes nos valores exibidos no menu do *display*, pode-se realizar um teste empírico, medindo a potência dissipada com equipamentos de precisão adequados à sua necessidade. Uma vez que tenham sido medidos os valores exatos de potência dissipada, altera-se no código a informação exibida nos submenus das funções *dispsubmenu* presentes no *switch(case)* principal do menu de navegação. Faz-se importante destacar que, pelo fato do módulo de cargas ter objetivos didáticos, se espera que o discente, ao utilizá-lo e realizar os testes e medições propostos em aula, verifique a diferença da potência de referência exibida no menu do *display* em relação à medida.

Existem outros equipamentos (máquinas elétricas e transformadores) no LABMAQ-EEL/UFSC, com potências diversas. O mesmo procedimento pode ser utilizado para se projetar novas cargas.

Por fim, espera-se que a tecnologia desenvolvida ofereça ao laboratório um ambiente mais agradável, seguro e eficiente para a aprendizagem de máquinas elétricas e transformadores, com reais ganhos de aprendizagem aos estudantes que realizam ensaios elétricos no laboratório.

REFERÊNCIAS

- AGRAWAL, D.; LEFF, H. S.; MENON, V. Efficiency and efficacy of incandescent lamps. *American Journal of Physics*, American Association of Physics Teachers, v. 64, n. 5, p. 649–654, 1996. Citado 2 vezes nas páginas 25 e 29.
- ALEXANDER, C. K.; SADIKU, M. N. *Fundamentos de circuitos elétricos*. [S.l.]: AMGH Editora, 2013. Citado 2 vezes nas páginas 43 e 44.
- ARDUINO. Arduino mega 2560 e lcd keypad shield. *Arduino Store*, 2021. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 20 de abr. 2021. Citado 2 vezes nas páginas 39 e 40.
- CAMPOS, R. A. F. Automação residencial utilizando arduino e aplicação web. 2014. Citado na página 38.
- CARDENUTO, F. M. Carcaça de encaixe do display e botões impressos em abs com impressora 3d. 2 fotografias. 2019. Citado na página 63.
- CHRISTO, E. d. S.; SOUZA, R. C. Uma abordagem estatística para a previsão de potência reativa em sistemas elétricos. *Pesquisa Operacional*, SciELO Brasil, v. 26, n. 2, p. 361–381, 2006. Citado 2 vezes nas páginas 20 e 21.
- COMITÊ DE DISTRIBUIÇÃO DE ENERGIA ELETRICA. *MANUAL DE ORIENTAÇÃO AOS CONSUMIDORES: Energia reativa excedente*. [S.l.], 2004. Citado 2 vezes nas páginas 20 e 21.
- COMRENT. *Resistive vs Reactive: Reasons to Choose Reactive Load Bank Testing Solutions*. [S.l.]: SI, 2019. v. 1. Citado na página 23.
- D-ROBOTICS. *LCD Keypad Shield*. [S.l.], 2011. Datasheet. Disponível em: <<https://datasheetspdf.com/pdf-file/746588/D-Robotics/DFR0009/1>>. Acesso em: 03 de maio 2021. Citado na página 57.
- DUARTE, A. Placa de split para alimentação dos reles. 1 fotografia. 2019. Citado na página 64.
- EBAY. Modulo de rele 16 canais. *Ebay*, 2021. Disponível em: <<https://www.ebay.com/>>. Acesso em: 20 de abr. 2021. Citado na página 39.
- ELECTRIC, S. O que significa a sigla ups? *SI*, SI, SI, n. SI, p. SI, 2019. Disponível em: <<https://www.se.com/br/pt/faqs/FA325755/>>. Acesso em: 24 de nov. 2020. Citado na página 23.
- ETHW. Early light bulbs. *Engineering and Technology History Wiki*, 2015. Disponível em: <https://ethw.org/Early_Light_Bulbs>. Acesso em: 14 de nov. 2020. Citado na página 24.
- FERRONI, E. et al. A plataforma arduino e suas aplicações. *Revista da UI_IPSantarém- Unidade de Investigação do Instituto Politécnico de Santarém*, 2015. Citado na página 45.

- FURFARI, F. A different kind of chemistry: a history of tungsten halogen lamps. *IEEE Industry Applications Magazine*, IEEE, v. 7, n. 6, p. 10–17, 2001. Citado 2 vezes nas páginas 25 e 29.
- GANDRA, A. Venda de lâmpadas incandescentes está proibida no país a partir de hoje. *Agência Brasil*, 2016. Disponível em: <<https://agenciabrasil.ebc.com.br/geral/noticia/2016-06/inmetro-inicia-fiscalizacao-no-varejo-de-lampadas-incandescentes-de-41-w-60-w>>. Acesso em: 17 de nov. 2020. Citado na página 28.
- GONZAGA, D. Circuito de interface para microcontroladores. *Embarcados*, 2015. Disponível em: <<https://www.embarcados.com.br/circuito-de-interface-para-microcontroladores/>>. Acesso em: 03 de mai. 2021. Citado na página 40.
- GRANT, J. Power quality issues - reactive power and power factor. *EM Magazine*, SI, SI, n. SI, p. SI, 2019. Disponível em: <<https://www.energymanagermagazine.co.uk/power-quality-issues-part-5-reactive-power-and-power-factor/>>. Acesso em: 20 de nov. 2020. Citado na página 21.
- GUEDES, D. B. *Linguagem de programação Python e Arduino como ferramenta para motivar estudantes iniciantes em programação*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2018. Citado na página 45.
- HALLIDAY, D.; RESNICK, R.; WALKER, J. Fundamentos de física, volume 2. *Lct edition*, 2011. Citado na página 22.
- HESSMANN, G. P. Sistema automatizado para imposição de faltas em uma máquina síncrona modificada. Florianópolis, SC, 2018. Citado na página 19.
- INSTITUTION, S. Preconditions to 20th century lamps. *American History*, 2020. Disponível em: <<https://americanhistory.si.edu/lighting/20thcent/prec20.htm>>. Acesso em: 14 de nov. 2020. Citado na página 25.
- LESLIE, R. P.; CONWAY, K. M. *Lighting Pattern Book for Homes*. [S.l.]: McGraw-Hill, 1996. v. 2. 109 p. Citado na página 24.
- LIGHT. *Tarifas e Tributos Grandes Clientes*. [S.l.: s.n.], S/A. Citado na página 20.
- MACISAAC, D.; KANNER, G.; ANDERSON, G. Basic physics of the incandescent lamp (lightbulb). *The physics teacher*, American Association of Physics Teachers, v. 37, n. 9, p. 520–525, 1999. Citado 3 vezes nas páginas 25, 26 e 27.
- MARAFON, C. et al. Benefícios do investimento em automação no processo de empacotamento de farinha de trigo. *Anais da Engenharia de Produção/ISSN 2594-4657*, v. 2, n. 1, p. 72–87, 2018. Citado na página 34.
- MARCAL, L. F.; GUIMARAES, M. P.; RESENDE, A. A. d. Automatização de uma termoformadora visando melhorias no processo produtivo de uma empresa fabricante de peças termoplásticas para o setor automobilístico. *XXXIII Encontro Nacional de Engenharia de Produção. Anais... Salvador*, 2013. Citado na página 34.
- MATULKA, R.; WOOD, D. The history of the light bulb. *U.S Department of Energy*, p. 5, 2013. Disponível em: <<https://www.energy.gov/articles/history-light-bulb>>. Acesso em: 24 de out. 2020. Citado na página 24.

- MAZZOLA, C. F. Layout interno e externo painel de cargas. 5 fotografias. 2019. Citado na página 62.
- MCROBERTS, M. *Beginning Arduino*. [S.l.]: Apress, 2011. Citado na página 37.
- MEHL, E. L. de M. Simulação de circuitos eletrônicos em computadores. 2010. Citado na página 56.
- MONK, S. *Programação com Arduino: começando com Sketches*. [S.l.]: Bookman Editora, 2013. Citado na página 38.
- POMILIO, J. A. Efeitos e causas de harmônicas no sistema de energia elétrica. *SI, SI, SI*, n. SI, p. SI, 1998. Disponível em: <<http://www.dsce.fee.unicamp.br/~antenor/htmlfile/harmo/fpcap4/cap4.html>>. Acesso em: 20 de nov. 2020. Citado na página 21.
- PRESTES, E.; RODRIGUES, L. M. Levantamento sobre a importância de aulas práticas para o ensino superior de engenharia. v. 8, 2020. Disponível em: <<https://periodicos.unipampa.edu.br/index.php/SIEPE/article/view/84976>>. Citado na página 18.
- QUEIROZ, P. V. S. de. *Mensuração do consumo de energia elétrica: algoritmo para detecção de potenciais usuários da termoacumulação como alternativa para deslocamento de carga*. [S.l.]: PUC-Rio, 2011. Citado na página 20.
- SANTOS, K. O. O software proteus e sua viabilidade no processo do ensino em circuitos elétricos. 2014. Citado na página 56.
- SHARP CORPORATION. *Photocoupler PC817x*. [S.l.], 2016. Datasheet. Disponível em: <<https://datasheet.octopart.com/PC817X1NIP1B-Sharp-datasheet-81842616.pdf>>. Acesso em: 27 de abril 2021. Citado na página 39.
- SILVA, L. D. d. *Análise de eficiência energética de lâmpadas incandescentes, fluorescentes e LEDs considerando as normas NBR 5410 e ISO 8995-1*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2016. Citado na página 24.
- SILVA, R. O. da; ARAUJO, W. M.; CAVALCANTE, M. M. Visão geral sobre microcontroladores e prototipagem com arduino. *TECNOLOGIAS EM PROJEÇÃO*, v. 10, n. 1, p. 36–46, 2019. Citado 2 vezes nas páginas 37 e 38.
- SILVEIRA, F. L. da. Potência elétrica em resistores não varia com o quadrado da tensão aplicada? o caso do filamento de tungstênio nas lâmpadas incandescentes. *Centro de Referência para Ensino de Física*, p. 5, 2015. Acesso em: 26 de out. 2020. Citado 2 vezes nas páginas 27 e 28.
- TORO, V. D. *Fundamentos de máquinas elétricas*. [S.l.]: Prentice-Hall do Brasil, 1994. Citado na página 30.
- TYPE TE SERIES. *High Power Wire Wound Resistor*. [S.l.], 2018. Datasheet. Disponível em: <<https://nz.element14.com/c/passive-components/resistors-fixed-value>>. Acesso em: 22 de out. 2020. Citado 2 vezes nas páginas 22 e 23.
- XAVIER, A. et al. *DETECTOR DE GÁS GLP*. [S.l.]: Brasil Escola, 2021. Citado na página 40.

APÊNDICE A – CÓDIGO FONTE MÓDULO DE CARGAS

```
// --- Bibliotecas Auxiliares ---  
#include <LiquidCrystal.h>  
//Biblioteca para o display LCD  
  
byte seta_cima[] = {  
    B00100,  
    B01110,  
    B10101,  
    B00100,  
    B00100,  
    B00100,  
    B00100,  
    B00100,  
    B00100  
};  
  
byte seta_baixo[] = {  
    B00100,  
    B00100,  
    B00100,  
    B00100,  
    B00100,  
    B10101,  
    B01110,  
    B00100  
};  
  
byte seta[] = {  
    B01000,  
    B00100,  
    B00010,  
    B11111,  
    B00010,  
    B00100,  
    B01000,  
    B00000  
};  
  
// --- Protótipo das Funções Auxiliares ---  
void keyboardRead();  
void dispMenu();  
void modoavaliacao();  
void dispmodoavaliacao();  
void dispsubmodoavaliacao();
```

```
void dispsubmenu_1();
void dispsubmenu_2();
void dispsubmenu_3();
void dispsubmenu_4();
void dispsubmenu_5();

void carga_1();
void carga_2();
void carga_3();
void carga_4();
void carga_5();
void carga_6();
void carga_7();
void carga_8();
void carga_9();
void carga_10();
void carga_11();
void carga_12();
void carga_13();
void carga_14();
void carga_15();
void carga_16();
void carga_17();
void carga_18();
void carga_19();
void carga_20();
void carga_21();
void carga_22();
void semcarga();

// --- Variáveis Globais ---
int adc_value = 0x00;
int x;
int y;

char menu = 0x01;
char submenu_1 = 0x01;
char submenu_2 = 0x01;
char submenu_3 = 0x01;
char submenu_4 = 0x01;
char submenu_5 = 0x01;
char submodoavaliacao = 0x01;

boolean right = 0x00, butt01 = 0x00,
up = 0x00, butt02 = 0x00,
```

```
    down    = 0x00, butt03 = 0x00,
    left    = 0x00, butt04 = 0x00,
    select  = 0x00, butt05 = 0x00;

// --- Hardware do LCD ---
LiquidCrystal disp(8,
                   9,
                   4,
                   5,
                   6,
                   7);

void setup()
{
    pinMode(26, INPUT);    //Pino de conexão para o switch do modo avaliação

    pinMode(27, OUTPUT);
    pinMode(28, OUTPUT);
    pinMode(29, OUTPUT);
    pinMode(30, OUTPUT);
    pinMode(31, OUTPUT);
    pinMode(32, OUTPUT);
    pinMode(33, OUTPUT);
    pinMode(34, OUTPUT);
    pinMode(35, OUTPUT);
    pinMode(36, OUTPUT);
    pinMode(37, OUTPUT);
    pinMode(38, OUTPUT);
    pinMode(39, OUTPUT);
    pinMode(40, OUTPUT);
    pinMode(41, OUTPUT);
    pinMode(42, OUTPUT);
    pinMode(43, OUTPUT);
    pinMode(44, OUTPUT);
    pinMode(45, OUTPUT);
    pinMode(46, OUTPUT);
    pinMode(47, OUTPUT);
    pinMode(48, OUTPUT);
    pinMode(49, OUTPUT);
    pinMode(50, OUTPUT);
    pinMode(51, OUTPUT);
    pinMode(52, OUTPUT);
    pinMode(53, OUTPUT);
```

```
disp.begin(16,2);
disp.createChar(0, seta_cima);
disp.createChar(1, seta_baixo);
disp.createChar(2, seta);

disp.setCursor(0,0);
disp.print (F(" Bem Vindo  "));
disp.setCursor(0,1);
disp.print(F("Painel de Carga"));

delay(200);

disp.clear();

if (digitalRead(26) == HIGH){

    modoavaliacao();

}

while (select == 0x00){

keyboardRead();

disp.setCursor(0,0);
disp.print(F("Pressione Select"));
disp.setCursor(0,1);
disp.print(F(" Para Iniciar  "));

}

} //end setup

void loop()
{

select = 0x00;

keyboardRead();
dispMenu();
changeMenu();

} //end loop
```

```
void keyboardRead()
{
    adc_value = analogRead(A0); //Variável para leitura recebe valor AD de A0

    // --- Testa se os botões foram pressionados ---
    // Se foi pressionado, seta a respectiva flag
    if (adc_value < 50)                butt01 = 0x01;
    else if (adc_value > 103 && adc_value < 200) butt02 = 0x01;
    else if (adc_value > 250 && adc_value < 380) butt03 = 0x01;
    else if (adc_value > 450 && adc_value < 550) butt04 = 0x01;
    else if (adc_value > 650 && adc_value < 800) butt05 = 0x01;

    // --- Testa se os botões foram liberados ---
    if (adc_value > 50 && butt01) //Botão right solto e flag butt01 setada?
    {
        //Sim...
        butt01 = 0x00;           //Limpa flag butt01
        right = 0x01;           //Seta flag right
    } //end right
    if (adc_value > 200 && butt02) //Botão up solto e flag butt02 setada?
    {
        //Sim...
        butt02 = 0x00;           //Limpa flag butt02
        up = 0x01;              //Seta flag up
    } //end up
    if (adc_value > 380 && butt03) //Botão down solto e flag butt03 setada?
    {
        //Sim...
        butt03 = 0x00;           //Limpa flag butt03
        down = 0x01;            //Seta flag down
    } //end down
    if (adc_value > 550 && butt04) //Botão left solto e flag butt04 setada?
    {
        //Sim...
        butt04 = 0x00;           //Limpa flag butt04
        left = 0x01;            //Seta flag left
    } //end left

    if (adc_value > 800 && butt05) //Botão left solto e flag butt04 setada?
    {
        //Sim...
        butt05 = 0x00;           //Limpa flag butt04
        select = 0x01;           //Seta flag left
    } //end select
}
```

```
} //end keyboardRead

void changeMenu()
{

    if(up == 0x01)
    {

        up = 0x00;

        menu--;

        if(menu < 0x01) menu = 0x01;

    }

    if(down == 0x01)
    {

        down = 0x00;

        menu++;

        if(menu > 0x05) menu = 0x05;

    }

}

void dispMenu()
{

    switch(menu)
    {

        case 0x01:

            disp.setCursor(15,0);
            disp.write((byte)0x01);

            disp.setCursor(0,0);
            disp.write((byte)0x02);

            disp.setCursor(1,0);
            disp.print(F("Monof. 110V  "));
            disp.setCursor(0,1);
            disp.print(F("Press. Select  "));

            if (select == 0x01){
```

```
        dispsubmenu_1();

    }
    break;

    case 0x02:

        disp.setCursor(15,0);
        disp.write((byte)0x00);

        disp.setCursor(15,1);
        disp.write((byte)0x01);

        disp.setCursor(0,0);
        disp.write((byte)0x02);

        disp.setCursor(1,0);
        disp.print(F("Monof. 220V  "));
        disp.setCursor(0,1);
        disp.print(F("Press. Select  "));

        if(select == 0x01){

            dispsubmenu_2();

        }

    break;

    case 0x03:

        disp.setCursor(15,0);
        disp.write((byte)0x00);

        disp.setCursor(15,1);
        disp.write((byte)0x01);

        disp.setCursor(0,0);
        disp.write((byte)0x02);

        disp.setCursor(1,0);
        disp.print(F("Trif. 220V Y  "));
        disp.setCursor(0,1);
        disp.print(F("Press. Select  "));

        if(select == 0x01){
```

```
        dispsubmenu_3();

    }

    break;

    case 0x04:

        disp.setCursor(15,0);
        disp.write((byte)0x00);

        disp.setCursor(15,1);
        disp.write((byte)0x01);

        disp.setCursor(0,0);
        disp.write((byte)0x02);

        disp.setCursor(1,0);
        disp.print(F("Trif. 380V Y "));
        disp.setCursor(0,1);
        disp.print(F("Press. Select "));

        if (select == 0x01){

            dispsubmenu_4();

        }

    break;

    case 0x05:

        disp.setCursor(15,0);
        disp.write((byte)0x00);

        disp.setCursor(0,0);
        disp.write((byte)0x02);

        disp.setCursor(1,0);
        disp.print(F("T. 220V Delta "));
        disp.setCursor(0,1);
        disp.print(F("Press. Select "));

        if (select == 0x01){
```

```
        dispsubmenu_5();

    }

    break;
}
}

void dispsubmenu_1()
{
    select = 0x00;
    x=1;

    while (x == 1){

        select = 0x00;

        if(right == 0x01)
        {
            right = 0x00;

            submenu_1++;

            if(submenu_1 > 0x17) submenu_1 = 0x17;
        }

        if(left == 0x01)
        {
            left = 0x00;

            submenu_1--;

            if(submenu_1 < 0x01) submenu_1 = 0x01;
        }

        switch(submenu_1)
        {

            case 0x01:

                keyboardRead();

                if (select == 0x01){
```

```
        x=0;
        up = 0x00;
        down = 0x00;

    }

    disp.setCursor(0,0);
    disp.print(F("Press. Select p/"));
    disp.setCursor(0,1);
    disp.print(F("Voltar   ou   >"));

    semcarga();

    break;

    case 0x02:

    keyboardRead();

    disp.setCursor(0,0);
    disp.write((byte)0x02);

    disp.setCursor(1,0);
    disp.print(F("Monofasico 110V"));
    disp.setCursor(0,1);
    disp.print(F("< P=15W           >"));

    carga_1();

    break;

    case 0x03:

    keyboardRead();

    disp.setCursor(0,0);
    disp.write((byte)0x02);

    disp.setCursor(1,0);
    disp.print(F("Monofasico 110V"));
    disp.setCursor(0,1);
    disp.print(F("< P=18,75W           >"));

    carga_2();
```

```
break;

case 0x04:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=25W          >"));

carga_3();

break;

case 0x05:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=37,5W          >"));

carga_4();

break;

case 0x06:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=62,5W          >"));

carga_5();

break;
```

```
    case 0x07:

        keyboardRead();

        disp.setCursor(0,0);
        disp.write((byte)0x02);

        disp.setCursor(1,0);
        disp.print(F("Monofasico 110V"));
        disp.setCursor(0,1);
        disp.print(F("< P=75W          >"));

        carga_6();

        break;

    case 0x08:

        keyboardRead();

        disp.setCursor(0,0);
        disp.write((byte)0x02);

        disp.setCursor(1,0);
        disp.print(F("Monofasico 110V"));
        disp.setCursor(0,1);
        disp.print(F("< P=90W          >"));

        carga_7();

        break;

    case 0x09:

        keyboardRead();

        disp.setCursor(0,0);
        disp.write((byte)0x02);

        disp.setCursor(1,0);
        disp.print(F("Monofasico 110V"));
        disp.setCursor(0,1);
        disp.print(F("< P=93,75W      >"));

        carga_8();

        break;
```

```
case 0x0A:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=100W      >"));

carga_9();

break;

case 0x0B:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=112,5W      >"));

carga_10();

break;

case 0x0C:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=137,5W      >"));

carga_11();
```

```
break;

case 0x0D:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=150W      >"));

carga_12();

break;

case 0x0E:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=168,75W    >"));

carga_13();

break;

case 0x0F:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=175W      >"));

carga_14();
```

```
break;

case 0x10:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=187,5W      >"));

carga_15();

break;

case 0x11:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=225W      >"));

carga_16();

break;

case 0x12:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
```

```
disp.setCursor(0,1);
disp.print(F("< P=250W      >"));

carga_17();

break;

case 0x13:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=262,5W      >"));

carga_18();

break;

case 0x14:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=300W      >"));

carga_19();

break;

case 0x15:

keyboardRead();

disp.setCursor(0,0);
```

```
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=337,5W      >"));

carga_20();

break;

case 0x16:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=375W      >"));

carga_21();

break;

case 0x17:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 110V"));
disp.setCursor(0,1);
disp.print(F("< P=400W      "));

carga_22();

break;
```

```
    }  
}  
}  
  
void dispsubmenu_2()  
{  
    select = 0x00;  
    x=1;  
  
    while (x == 1){  
  
        select = 0x00;  
  
        if(right == 0x01)  
        {  
            right = 0x00;  
  
            submenu_2 ++;  
  
            if(submenu_2 > 0x17) submenu_2 = 0x17;  
        }  
  
        if(left == 0x01)  
        {  
            left = 0x00;  
  
            submenu_2 --;  
  
            if(submenu_2 < 0x01) submenu_2 = 0x01;  
        }  
  
        switch(submenu_2)  
        {  
  
            case 0x01:  
  
                keyboardRead();  
  
                if (select == 0x01){  
  
                    x=0;  
                    up = 0x00;
```

```
        down = 0x00;

    }

    disp.setCursor(0,0);
    disp.print(F("Press. Select p/"));
    disp.setCursor(0,1);
    disp.print(F("Voltar   ou   >"));

    semcarga();

    break;

    case 0x02:

    keyboardRead();

    disp.setCursor(0,0);
    disp.write((byte)0x02);

    disp.setCursor(1,0);
    disp.print(F("Monofasico 220V"));
    disp.setCursor(0,1);
    disp.print(F("< P=60W           >"));

    carga_1();

    break;

    case 0x03:

    keyboardRead();

    disp.setCursor(0,0);
    disp.write((byte)0x02);

    disp.setCursor(1,0);
    disp.print(F("Monofasico 220V"));
    disp.setCursor(0,1);
    disp.print(F("< P=75W           >"));

    carga_2();

    break;

    case 0x04:

    keyboardRead();
```

```
disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=100W      >"));

carga_3();

break;

case 0x05:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=150W      >"));

carga_4();

break;

case 0x06:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=250W      >"));

carga_5();

break;

case 0x07:

keyboardRead();
```

```
disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=300W      >"));

carga_6();

break;

case 0x08:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=360W      >"));

carga_7();

break;

case 0x09:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=375W      >"));

carga_8();

break;

case 0x0A:

keyboardRead();

disp.setCursor(0,0);
```

```
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=400W      >"));

carga_9();

break;

case 0x0B:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=450W      >"));

carga_10();

break;

case 0x0C:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=550W      >"));

carga_11();

break;

case 0x0D:

keyboardRead();
```

```
disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=600W      >"));

carga_12();

break;

case 0x0E:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=675W      >"));

carga_13();

break;

case 0x0F:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=700W      >"));

carga_14();

break;

case 0x10:

keyboardRead();
```

```
disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=750W      >"));

carga_15();

break;

case 0x11:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=900W      >"));

carga_16();

break;

case 0x12:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=1000W     >"));

carga_17();

break;

case 0x13:

keyboardRead();

disp.setCursor(0,0);
```

```
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=1050W      >"));

carga_18();

break;

case 0x14:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=1200W      >"));

carga_19();

break;

case 0x15:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Monofasico 220V"));
disp.setCursor(0,1);
disp.print(F("< P=1350W      >"));

carga_20();

break;

case 0x16:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);
```

```
    disp.setCursor(1,0);
    disp.print(F("Monofasico 220V"));
    disp.setCursor(0,1);
    disp.print(F("< P=1500W      >"));

    carga_21();

    break;

    case 0x17:

    keyboardRead();

    disp.setCursor(0,0);
    disp.write((byte)0x02);

    disp.setCursor(1,0);
    disp.print(F("Monofasico 220V"));
    disp.setCursor(0,1);
    disp.print(F("< P=1800W      "));

    carga_22();

    break;

}

}

}

void dispsubmenu_3()
{
    select = 0x00;
    x=1;

    while (x == 1){

    select = 0x00;

    if(right == 0x01)
    {
        right = 0x00;

        submenu_3 ++;
    }
}
}
```

```
    if(submenu_3 > 0x17) submenu_3 = 0x17;
}

if(left == 0x01)
{
    left = 0x00;

    submenu_3 --;

    if(submenu_3 < 0x01) submenu_3 = 0x01;
}

switch(submenu_3)
{

    case 0x01:

        keyboardRead();

        if (select == 0x01){

            x=0;
            up = 0x00;
            down = 0x00;

        }

        disp.setCursor(0,0);
        disp.print(F("Press. Select p/"));
        disp.setCursor(0,1);
        disp.print(F("Voltar   ou   >"));

        semcarga();

        break;

    case 0x02:

        keyboardRead();

        disp.setCursor(0,0);
        disp.write((byte)0x02);

        disp.setCursor(1,0);
```

```
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=45W      >"));

carga_1();

break;

case 0x03:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=56,3W      >"));

carga_2();

break;

case 0x04:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=75W      >"));

carga_3();

break;

case 0x05:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
```

```
disp.setCursor(0,1);
disp.print(F("< P=112,5W    >"));

carga_4();

break;

case 0x06:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=187,5W    >"));

carga_5();

break;

case 0x07:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=225W    >"));

carga_6();

break;

case 0x08:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
```

```
disp.print(F("< P=270W      >"));

carga_7();

break;

case 0x09:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=281,3W      >"));

carga_8();

break;

case 0x0A:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=300W      >"));

carga_9();

break;

case 0x0B:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
```

```
disp.print(F("< P=337,5W   >"));

carga_10();

break;

case 0x0C:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y   "));
disp.setCursor(0,1);
disp.print(F("< P=412,5W   >"));

carga_11();

break;

case 0x0D:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y   "));
disp.setCursor(0,1);
disp.print(F("< P=450W   >"));

carga_12();

break;

case 0x0E:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y   "));
disp.setCursor(0,1);
```

```
disp.print(F("< P=506,3W   >"));

carga_13();

break;

case 0x0F:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y   "));
disp.setCursor(0,1);
disp.print(F("< P=525W   >"));

carga_14();

break;

case 0x10:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y   "));
disp.setCursor(0,1);
disp.print(F("< P=562,5W   >"));

carga_15();

break;

case 0x11:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y   "));
disp.setCursor(0,1);
```

```
disp.print(F("< P=675W      >"));

carga_16();

break;

case 0x12:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=750W      >"));

carga_17();

break;

case 0x13:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=787,5W      >"));

carga_18();

break;

case 0x14:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=900W      >"));
```

```
carga_19();

break;

case 0x15:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=1012,5W  >"));

carga_20();

break;

case 0x16:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=1125W  >"));

carga_21();

break;

case 0x17:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 220V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=1350W  "));
```

```
        carga_22();

        break;

    }

}

}

void dispsubmenu_4()
{
    select = 0x00;
    x=1;

    while (x == 1){

        select = 0x00;

        if(right == 0x01)
        {
            right = 0x00;

            submenu_4 ++;

            if(submenu_4 > 0x17) submenu_4 = 0x17;
        }

        if(left == 0x01)
        {
            left = 0x00;

            submenu_4 --;

            if(submenu_4 < 0x01) submenu_4 = 0x01;
        }

        switch(submenu_4)
        {

            case 0x01:

                keyboardRead();
```

```
if (select == 0x01){

    x=0;
    up = 0x00;
    down = 0x00;

}

disp.setCursor(0,0);
disp.print(F("Press. Select p/"));
disp.setCursor(0,1);
disp.print(F("Voltar ou >"));

semcarga();

break;

case 0x02:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y "));
disp.setCursor(0,1);
disp.print(F("< P=180W >"));

carga_1();

break;

case 0x03:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y "));
disp.setCursor(0,1);
disp.print(F("< P=225W >"));

carga_2();
```

```
break;

case 0x04:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=300W      >"));

carga_3();

break;

case 0x05:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=450W      >"));

carga_4();

break;

case 0x06:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=750W      >"));

carga_5();

break;
```

```
case 0x07:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=900W      >"));

carga_6();

break;

case 0x08:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=1080W     >"));

carga_7();

break;

case 0x09:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=1125W     >"));

carga_8();

break;
```

```
case 0x0A:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=1200W      >"));

carga_9();

break;

case 0x0B:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=1350W      >"));

carga_10();

break;

case 0x0C:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=1650W      >"));

carga_11();

break;
```

```
case 0x0D:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=1800W      >"));

carga_12();

break;

case 0x0E:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=2025W      >"));

carga_13();

break;

case 0x0F:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=2100W      >"));

carga_14();

break;
```

```
case 0x10:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=2250W      >"));

carga_15();

break;

case 0x11:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=2700W      >"));

carga_16();

break;

case 0x12:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=3000W      >"));

carga_17();

break;
```

```
case 0x13:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=3150W  >"));

carga_18();

break;

case 0x14:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=3600W  >"));

carga_19();

break;

case 0x15:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Trif. 380V Y  "));
disp.setCursor(0,1);
disp.print(F("< P=4050W  >"));

carga_20();

break;

case 0x16:
```

```
    keyboardRead();

    disp.setCursor(0,0);
    disp.write((byte)0x02);

    disp.setCursor(1,0);
    disp.print(F("Trif. 380V Y  "));
    disp.setCursor(0,1);
    disp.print(F("< P=4500W      >"));

    carga_21();

    break;

    case 0x17:

    keyboardRead();

    disp.setCursor(0,0);
    disp.write((byte)0x02);

    disp.setCursor(1,0);
    disp.print(F("Trif. 380V Y  "));
    disp.setCursor(0,1);
    disp.print(F("< P=5400W      "));

    carga_22();

    break;

}
}

void dispsubmenu_5()
{
    select = 0x00;
    x=1;

    while (x == 1){

    select = 0x00;

    if(right == 0x01)
    {
        right = 0x00;
```

```
    submenu_5 ++;

    if(submenu_5 > 0x17) submenu_5 = 0x17;
}

if(left == 0x01)
{
    left = 0x00;

    submenu_5 --;

    if(submenu_5 < 0x01) submenu_5 = 0x01;
}

switch(submenu_5)
{

    case 0x01:

        keyboardRead();

        if (select == 0x01){

            x=0;
            up = 0x00;
            down = 0x00;

        }

        disp.setCursor(0,0);
        disp.print(F("Press. Select p/"));
        disp.setCursor(0,1);
        disp.print(F("Voltar   ou   >"));

        semcarga();

        break;

    case 0x02:

        keyboardRead();

        disp.setCursor(0,0);
```

```
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=180W      >"));

carga_1();

break;

case 0x03:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=225W      >"));

carga_2();

break;

case 0x04:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=300W      >"));

carga_3();

break;

case 0x05:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);
```

```
disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=450W      >"));

carga_4();

break;

case 0x06:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=750W      >"));

carga_5();

break;

case 0x07:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=900W      >"));

carga_6();

break;

case 0x08:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);
```

```
disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=1080W      >"));

carga_7();

break;

case 0x09:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=1125W      >"));

carga_8();

break;

case 0x0A:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=1200W      >"));

carga_9();

break;

case 0x0B:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);
```

```
disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=1350W      >"));

carga_10();

break;

case 0x0C:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=1650W      >"));

carga_11();

break;

case 0x0D:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=1800W      >"));

carga_12();

break;

case 0x0E:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);
```

```
disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=2025W      >"));

carga_13();

break;

case 0x0F:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=2100W      >"));

carga_14();

break;

case 0x10:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=2250W      >"));

carga_15();

break;

case 0x11:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);
```

```
disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=2700W      >"));

carga_16();

break;

case 0x12:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=3000W      >"));

carga_17();

break;

case 0x13:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=3150W      >"));

carga_18();

break;

case 0x14:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
```

```
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=3600W      >"));

carga_19();

break;

case 0x15:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=4050W      >"));

carga_20();

break;

case 0x16:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=4500W      >"));

carga_21();

break;

case 0x17:

keyboardRead();

disp.setCursor(0,0);
disp.write((byte)0x02);

disp.setCursor(1,0);
```

```
disp.print(F("Tri. 220V Delta"));
disp.setCursor(0,1);
disp.print(F("< P=5400W      "));

carga_22();

break;

}
}
}

void modoavaliacao()
{

    y = 0x01;

    while(y == 0x01){

        keyboardRead();
        dispmodoavaliacao();
    }

}

void dispmodoavaliacao()
{

    disp.setCursor(0,0);
    disp.write((byte)0x02);

    disp.setCursor(1,0);
    disp.print(" Modo Avaliacao");
    dispsubmodoavaliacao();

}

void dispsubmodoavaliacao()
{

if(right == 0x01)
{
    right = 0x00;
```

```
    submodoavaliacao++;

    if(submodoavaliacao > 0x17) submodoavaliacao = 0x17;
}

if(left == 0x01)
{
    left = 0x00;

    submodoavaliacao--;

    if(submodoavaliacao < 0x01) submodoavaliacao = 0x01;
}

switch(submodoavaliacao)
{
    case 0x01:

        disp.setCursor(0,1);
        disp.print("  Sem Carga  >");

        semcarga();

        break;

    case 0x02:

        disp.setCursor(0,1);
        disp.print("<  Carga 1  >");

        carga_1();

        break;

    case 0x03:

        disp.setCursor(0,1);
        disp.print("<  Carga 2  >");

        carga_2();

        break;

    case 0x04:
```

```
disp.setCursor(0,1);
disp.print("<  Carga 3  >");

carga_3();

break;

case 0x05:

disp.setCursor(0,1);
disp.print("<  Carga 4  >");

carga_4();

break;

case 0x06:

disp.setCursor(0,1);
disp.print("<  Carga 5  >");

carga_5();

break;

case 0x07:

disp.setCursor(0,1);
disp.print("<  Carga 6  >");

carga_6();

break;

case 0x08:

disp.setCursor(0,1);
disp.print("<  Carga 7  >");

carga_7();

break;

case 0x09:

disp.setCursor(0,1);
disp.print("<  Carga 8  >");
```

```
carga_8();

break;

case 0x0A:

disp.setCursor(0,1);
disp.print("< Carga 9 >");

carga_9();

break;

case 0x0B:

disp.setCursor(0,1);
disp.print("< Carga 10 >");

carga_10();

break;

case 0x0C:

disp.setCursor(0,1);
disp.print("< Carga 11 >");

carga_11();

break;

case 0x0D:

disp.setCursor(0,1);
disp.print("< Carga 12 >");

carga_12();

break;

case 0x0E:

disp.setCursor(0,1);
disp.print("< Carga 13 >");

carga_13();

break;
```

```
    case 0x0F:

disp.setCursor(0,1);
disp.print("<  Carga 14  >");

carga_14();

break;

    case 0x10:

disp.setCursor(0,1);
disp.print("<  Carga 15  >");

carga_15();

break;

    case 0x11:

disp.setCursor(0,1);
disp.print("<  Carga 16  >");

carga_16();

break;

    case 0x12:

disp.setCursor(0,1);
disp.print("<  Carga 17  >");

carga_17();

break;

    case 0x13:

disp.setCursor(0,1);
disp.print("<  Carga 18  >");

carga_18();

break;

    case 0x14:

disp.setCursor(0,1);
```

```
    disp.print("<  Carga 19  >");

    carga_19();

    break;

    case 0x15:

    disp.setCursor(0,1);
    disp.print("<  Carga 20  >");

    carga_20();

    break;

    case 0x16:

    disp.setCursor(0,1);
    disp.print("<  Carga 21  >");

    carga_21();

    break;

    case 0x17:

    disp.setCursor(0,1);
    disp.print("<  Carga 22  ");

    carga_22();

    break;

}

}

void carga_1 ()
{

    digitalWrite(27 , HIGH);
    digitalWrite(28 , LOW);
    digitalWrite(29 , LOW);
    digitalWrite(30 , LOW);
    digitalWrite(31 , LOW);
    digitalWrite(32 , LOW);
    digitalWrite(33 , LOW);
```

```
digitalWrite(34 , LOW);
digitalWrite(35 , LOW);
digitalWrite(36 , HIGH);
digitalWrite(37 , LOW);
digitalWrite(38 , LOW);
digitalWrite(39 , LOW);
digitalWrite(40 , LOW);
digitalWrite(41 , LOW);
digitalWrite(42 , LOW);
digitalWrite(43 , LOW);
digitalWrite(44 , LOW);
digitalWrite(45 , HIGH);
digitalWrite(46 , LOW);
digitalWrite(47 , LOW);
digitalWrite(48 , LOW);
digitalWrite(49 , LOW);
digitalWrite(50 , LOW);
digitalWrite(51 , LOW);
digitalWrite(52 , LOW);
digitalWrite(53 , LOW);
}

void carga_2 ()
{

digitalWrite(27 , HIGH);
digitalWrite(28 , HIGH);
digitalWrite(29 , LOW);
digitalWrite(30 , LOW);
digitalWrite(31 , LOW);
digitalWrite(32 , LOW);
digitalWrite(33 , LOW);
digitalWrite(34 , LOW);
digitalWrite(35 , LOW);
digitalWrite(36 , HIGH);
digitalWrite(37 , HIGH);
digitalWrite(38 , LOW);
digitalWrite(39 , LOW);
digitalWrite(40 , LOW);
digitalWrite(41 , LOW);
digitalWrite(42 , LOW);
digitalWrite(43 , LOW);
digitalWrite(44 , LOW);
digitalWrite(45 , HIGH);
digitalWrite(46 , HIGH);
digitalWrite(47 , LOW);
digitalWrite(48 , LOW);
digitalWrite(49 , LOW);
```

```
        digitalWrite(50 , LOW);
        digitalWrite(51 , LOW);
        digitalWrite(52 , LOW);
        digitalWrite(53 , LOW);
    }

    void carga_3()
    {
        digitalWrite(27 , HIGH);
        digitalWrite(28 , LOW);
        digitalWrite(29 , HIGH);
        digitalWrite(30 , LOW);
        digitalWrite(31 , LOW);
        digitalWrite(32 , LOW);
        digitalWrite(33 , LOW);
        digitalWrite(34 , LOW);
        digitalWrite(35 , LOW);
        digitalWrite(36 , HIGH);
        digitalWrite(37 , LOW);
        digitalWrite(38 , HIGH);
        digitalWrite(39 , LOW);
        digitalWrite(40 , LOW);
        digitalWrite(41 , LOW);
        digitalWrite(42 , LOW);
        digitalWrite(43 , LOW);
        digitalWrite(44 , LOW);
        digitalWrite(45 , HIGH);
        digitalWrite(46 , LOW);
        digitalWrite(47 , HIGH);
        digitalWrite(48 , LOW);
        digitalWrite(49 , LOW);
        digitalWrite(50 , LOW);
        digitalWrite(51 , LOW);
        digitalWrite(52 , LOW);
        digitalWrite(53 , LOW);
    }

    void carga_4()
    {
        digitalWrite(27 , HIGH);
        digitalWrite(28 , LOW);
        digitalWrite(29 , LOW);
        digitalWrite(30 , HIGH);
        digitalWrite(31 , LOW);
        digitalWrite(32 , LOW);
        digitalWrite(33 , LOW);
        digitalWrite(34 , LOW);
        digitalWrite(35 , LOW);
    }
```

```
digitalWrite(36 , HIGH);
digitalWrite(37 , LOW);
digitalWrite(38 , LOW);
digitalWrite(39 , HIGH);
digitalWrite(40 , LOW);
digitalWrite(41 , LOW);
digitalWrite(42 , LOW);
digitalWrite(43 , LOW);
digitalWrite(44 , LOW);
digitalWrite(45 , HIGH);
digitalWrite(46 , LOW);
digitalWrite(47 , LOW);
digitalWrite(48 , HIGH);
digitalWrite(49 , LOW);
digitalWrite(50 , LOW);
digitalWrite(51 , LOW);
digitalWrite(52 , LOW);
digitalWrite(53 , LOW);
}

void carga_5()
{
digitalWrite(27 , LOW);
digitalWrite(28 , LOW);
digitalWrite(29 , LOW);
digitalWrite(30 , HIGH);
digitalWrite(31 , LOW);
digitalWrite(32 , LOW);
digitalWrite(33 , LOW);
digitalWrite(34 , LOW);
digitalWrite(35 , LOW);
digitalWrite(36 , LOW);
digitalWrite(37 , LOW);
digitalWrite(38 , LOW);
digitalWrite(39 , HIGH);
digitalWrite(40 , LOW);
digitalWrite(41 , LOW);
digitalWrite(42 , LOW);
digitalWrite(43 , LOW);
digitalWrite(44 , LOW);
digitalWrite(45 , LOW);
digitalWrite(46 , LOW);
digitalWrite(47 , LOW);
digitalWrite(48 , HIGH);
digitalWrite(49 , LOW);
digitalWrite(50 , LOW);
digitalWrite(51 , LOW);
digitalWrite(52 , LOW);
digitalWrite(53 , LOW);
```

```
}

void carga_6()
{
    digitalWrite(27 , LOW);
    digitalWrite(28, LOW);
    digitalWrite(29, LOW);
    digitalWrite(30, LOW);
    digitalWrite(31 , LOW);
    digitalWrite(32 , LOW);
    digitalWrite(33 , LOW);
    digitalWrite(34 , LOW);
    digitalWrite(35 , HIGH);
    digitalWrite(36 , LOW);
    digitalWrite(37, LOW);
    digitalWrite(38, LOW);
    digitalWrite(39, LOW);
    digitalWrite(40 , LOW);
    digitalWrite(41 , LOW);
    digitalWrite(42 , LOW);
    digitalWrite(43 , LOW);
    digitalWrite(44 , HIGH);
    digitalWrite(45 , LOW);
    digitalWrite(46, LOW);
    digitalWrite(47, LOW);
    digitalWrite(48, LOW);
    digitalWrite(49 , LOW);
    digitalWrite(50 , LOW);
    digitalWrite(51 , LOW);
    digitalWrite(52 , LOW);
    digitalWrite(53 , HIGH);
}

void carga_7()
{
    digitalWrite(27 , HIGH);
    digitalWrite(28, LOW);
    digitalWrite(29, LOW);
    digitalWrite(30, LOW);
    digitalWrite(31 , LOW);
    digitalWrite(32 , LOW);
    digitalWrite(33 , LOW);
    digitalWrite(34 , LOW);
    digitalWrite(35 , HIGH);
    digitalWrite(36 , HIGH);
    digitalWrite(37, LOW);
    digitalWrite(38, LOW);
    digitalWrite(39, LOW);
    digitalWrite(40 , LOW);
}
```

```
    digitalWrite(41 , LOW);
    digitalWrite(42 , LOW);
    digitalWrite(43 , LOW);
    digitalWrite(44 , HIGH);
    digitalWrite(45 , HIGH);
    digitalWrite(46, LOW);
    digitalWrite(47, LOW);
    digitalWrite(48, LOW);
    digitalWrite(49 , LOW);
    digitalWrite(50 , LOW);
    digitalWrite(51 , LOW);
    digitalWrite(52 , LOW);
    digitalWrite(53 , HIGH);
}

void carga_8()
{
    digitalWrite(27 , HIGH);
    digitalWrite(28, HIGH);
    digitalWrite(29, LOW);
    digitalWrite(30, LOW);
    digitalWrite(31 , LOW);
    digitalWrite(32 , LOW);
    digitalWrite(33 , LOW);
    digitalWrite(34 , LOW);
    digitalWrite(35 , HIGH);
    digitalWrite(36 , HIGH);
    digitalWrite(37, HIGH);
    digitalWrite(38, LOW);
    digitalWrite(39, LOW);
    digitalWrite(40 , LOW);
    digitalWrite(41 , LOW);
    digitalWrite(42 , LOW);
    digitalWrite(43 , LOW);
    digitalWrite(44 , HIGH);
    digitalWrite(45 , HIGH);
    digitalWrite(46, HIGH);
    digitalWrite(47, LOW);
    digitalWrite(48, LOW);
    digitalWrite(49 , LOW);
    digitalWrite(50 , LOW);
    digitalWrite(51 , LOW);
    digitalWrite(52 , LOW);
    digitalWrite(53 , HIGH);
}

void carga_9()
{
    digitalWrite(27 , HIGH);
```

```
digitalWrite(28 , LOW);
digitalWrite(29 , HIGH);
digitalWrite(30 , LOW);
digitalWrite(31 , LOW);
digitalWrite(32 , LOW);
digitalWrite(33 , LOW);
digitalWrite(34 , LOW);
digitalWrite(35 , HIGH);
digitalWrite(36 , HIGH);
digitalWrite(37 , LOW);
digitalWrite(38 , HIGH);
digitalWrite(39 , LOW);
digitalWrite(40 , LOW);
digitalWrite(41 , LOW);
digitalWrite(42 , LOW);
digitalWrite(43 , LOW);
digitalWrite(44 , HIGH);
digitalWrite(45 , HIGH);
digitalWrite(46 , LOW);
digitalWrite(47 , HIGH);
digitalWrite(48 , LOW);
digitalWrite(49 , LOW);
digitalWrite(50 , LOW);
digitalWrite(51 , LOW);
digitalWrite(52 , LOW);
digitalWrite(53 , HIGH);
}

void carga_10()
{
digitalWrite(27 , HIGH);
digitalWrite(28 , LOW);
digitalWrite(29 , LOW);
digitalWrite(30 , HIGH);
digitalWrite(31 , LOW);
digitalWrite(32 , LOW);
digitalWrite(33 , LOW);
digitalWrite(34 , LOW);
digitalWrite(35 , HIGH);
digitalWrite(36 , HIGH);
digitalWrite(37 , LOW);
digitalWrite(38 , LOW);
digitalWrite(39 , HIGH);
digitalWrite(40 , LOW);
digitalWrite(41 , LOW);
digitalWrite(42 , LOW);
digitalWrite(43 , LOW);
digitalWrite(44 , HIGH);
digitalWrite(45 , HIGH);
```

```
    digitalWrite(46 , LOW);
    digitalWrite(47 , LOW);
    digitalWrite(48 , HIGH);
    digitalWrite(49 , LOW);
    digitalWrite(50 , LOW);
    digitalWrite(51 , LOW);
    digitalWrite(52 , LOW);
    digitalWrite(53 , HIGH);
}

void carga_11()
{
    digitalWrite(27 , LOW);
    digitalWrite(28, LOW);
    digitalWrite(29 , HIGH);
    digitalWrite(30 , LOW);
    digitalWrite(31 , LOW);
    digitalWrite(32 , LOW);
    digitalWrite(33 , LOW);
    digitalWrite(34 , LOW);
    digitalWrite(35 , HIGH);
    digitalWrite(36 , LOW);
    digitalWrite(37, LOW);
    digitalWrite(38 , HIGH);
    digitalWrite(39 , LOW);
    digitalWrite(40 , LOW);
    digitalWrite(41 , LOW);
    digitalWrite(42 , LOW);
    digitalWrite(43 , LOW);
    digitalWrite(44 , HIGH);
    digitalWrite(45 , LOW);
    digitalWrite(46, LOW);
    digitalWrite(47 , HIGH);
    digitalWrite(48 , LOW);
    digitalWrite(49 , LOW);
    digitalWrite(50 , LOW);
    digitalWrite(51 , LOW);
    digitalWrite(52 , LOW);
    digitalWrite(53 , HIGH);
}

void carga_12()
{
    digitalWrite(27 , HIGH);
    digitalWrite(28, LOW);
    digitalWrite(29, LOW);
    digitalWrite(30, LOW);
    digitalWrite(31 , HIGH);
    digitalWrite(32 , LOW);
```

```
digitalWrite(33 , LOW);
digitalWrite(34 , LOW);
digitalWrite(35 , HIGH);
digitalWrite(36 , HIGH);
digitalWrite(37, LOW);
digitalWrite(38, LOW);
digitalWrite(39, LOW);
digitalWrite(40 , HIGH);
digitalWrite(41 , LOW);
digitalWrite(42 , LOW);
digitalWrite(43 , LOW);
digitalWrite(44 , HIGH);
digitalWrite(45 , HIGH);
digitalWrite(46, LOW);
digitalWrite(47, LOW);
digitalWrite(48, LOW);
digitalWrite(49 , HIGH);
digitalWrite(50 , LOW);
digitalWrite(51 , LOW);
digitalWrite(52, LOW);
digitalWrite(53 , HIGH);
}

void carga_13()
{
digitalWrite(27 , LOW);
digitalWrite(28 , LOW);
digitalWrite(29, LOW);
digitalWrite(30, LOW);
digitalWrite(31 , HIGH);
digitalWrite(32 , LOW);
digitalWrite(33 , LOW);
digitalWrite(34 , LOW);
digitalWrite(35 , HIGH);
digitalWrite(36 , LOW);
digitalWrite(37 , LOW);
digitalWrite(38, LOW);
digitalWrite(39, LOW);
digitalWrite(40 , HIGH);
digitalWrite(41 , LOW);
digitalWrite(42 , LOW);
digitalWrite(43 , LOW);
digitalWrite(44 , HIGH);
digitalWrite(45 , LOW);
digitalWrite(46 , LOW);
digitalWrite(47, LOW);
digitalWrite(48, LOW);
digitalWrite(49 , HIGH);
digitalWrite(50, LOW);
```

```
        digitalWrite(51 , LOW);
        digitalWrite(52 , LOW);
        digitalWrite(53 , HIGH);
    }

    void carga_14()
    {
        digitalWrite(27 , LOW);
        digitalWrite(28, LOW);
        digitalWrite(29, LOW);
        digitalWrite(30, HIGH);
        digitalWrite(31 , HIGH);
        digitalWrite(32 , LOW);
        digitalWrite(33 , LOW);
        digitalWrite(34 , LOW);
        digitalWrite(35 , HIGH);
        digitalWrite(36 , LOW);
        digitalWrite(37, LOW);
        digitalWrite(38, LOW);
        digitalWrite(39, HIGH);
        digitalWrite(40 , HIGH);
        digitalWrite(41 , LOW);
        digitalWrite(42 , LOW);
        digitalWrite(43 , LOW);
        digitalWrite(44 , HIGH);
        digitalWrite(45 , LOW);
        digitalWrite(46, LOW);
        digitalWrite(47, LOW);
        digitalWrite(48, HIGH);
        digitalWrite(49 , HIGH);
        digitalWrite(50 , LOW);
        digitalWrite(51 , LOW);
        digitalWrite(52 , LOW);
        digitalWrite(53 , HIGH);
    }

    void carga_15()
    {
        digitalWrite(27 , LOW);
        digitalWrite(28, LOW);
        digitalWrite(29, HIGH);
        digitalWrite(30, LOW);
        digitalWrite(31 , HIGH);
        digitalWrite(32 , LOW);
        digitalWrite(33 , LOW);
        digitalWrite(34 , LOW);
        digitalWrite(35 , HIGH);
        digitalWrite(36 , LOW);
```

```
digitalWrite(37, LOW);
digitalWrite(38, HIGH);
digitalWrite(39, LOW);
digitalWrite(40, HIGH);
digitalWrite(41, LOW);
digitalWrite(42, LOW);
digitalWrite(43, LOW);
digitalWrite(44, HIGH);
digitalWrite(45, LOW);
digitalWrite(46, LOW);
digitalWrite(47, HIGH);
digitalWrite(48, LOW);
digitalWrite(49, HIGH);
digitalWrite(50, LOW);
digitalWrite(51, LOW);
digitalWrite(52, LOW);
digitalWrite(53, HIGH);
}

void carga_16()
{
digitalWrite(27, LOW);
digitalWrite(28, LOW);
digitalWrite(29, LOW);
digitalWrite(30, HIGH);
digitalWrite(31, HIGH);
digitalWrite(32, LOW);
digitalWrite(33, LOW);
digitalWrite(34, HIGH);
digitalWrite(35, HIGH);
digitalWrite(36, LOW);
digitalWrite(37, LOW);
digitalWrite(38, LOW);
digitalWrite(39, HIGH);
digitalWrite(40, HIGH);
digitalWrite(41, LOW);
digitalWrite(42, LOW);
digitalWrite(43, HIGH);
digitalWrite(44, HIGH);
digitalWrite(45, LOW);
digitalWrite(46, LOW);
digitalWrite(47, LOW);
digitalWrite(48, HIGH);
digitalWrite(49, HIGH);
digitalWrite(50, LOW);
digitalWrite(51, LOW);
digitalWrite(52, HIGH);
digitalWrite(53, HIGH);
}
```

```
void carga_17()
{
    digitalWrite(27 , HIGH);
    digitalWrite(28, LOW);
    digitalWrite(29, LOW);
    digitalWrite(30, HIGH);
    digitalWrite(31 , HIGH);
    digitalWrite(32 , LOW);
    digitalWrite(33 , LOW);
    digitalWrite(34 , HIGH);
    digitalWrite(35 , HIGH);
    digitalWrite(36 , HIGH);
    digitalWrite(37, LOW);
    digitalWrite(38, LOW);
    digitalWrite(39, HIGH);
    digitalWrite(40 , HIGH);
    digitalWrite(41 , LOW);
    digitalWrite(42 , LOW);
    digitalWrite(43 , HIGH);
    digitalWrite(44 , HIGH);
    digitalWrite(45 , HIGH);
    digitalWrite(46, LOW);
    digitalWrite(47, LOW);
    digitalWrite(48, HIGH);
    digitalWrite(49 , HIGH);
    digitalWrite(50 , LOW);
    digitalWrite(51 , LOW);
    digitalWrite(52 , HIGH);
    digitalWrite(53 , HIGH);
}
```

```
void carga_18()
{
    digitalWrite(27 , HIGH);
    digitalWrite(28, LOW);
    digitalWrite(29, HIGH);
    digitalWrite(30 , HIGH);
    digitalWrite(31 , HIGH);
    digitalWrite(32 , LOW);
    digitalWrite(33 , HIGH);
    digitalWrite(34 , HIGH);
    digitalWrite(35 , HIGH);
    digitalWrite(36 , HIGH);
    digitalWrite(37, LOW);
    digitalWrite(38, HIGH);
    digitalWrite(39 , HIGH);
    digitalWrite(40 , HIGH);
    digitalWrite(41 , LOW);
```

```
digitalWrite(42 , HIGH);
digitalWrite(43 , HIGH);
digitalWrite(44 , HIGH);
digitalWrite(45 , HIGH);
digitalWrite(46, LOW);
digitalWrite(47, HIGH);
digitalWrite(48 , HIGH);
digitalWrite(49 , HIGH);
digitalWrite(50 , LOW);
digitalWrite(51 , HIGH);
digitalWrite(52 , HIGH);
digitalWrite(53 , HIGH);
}

void carga_19()
{
digitalWrite(27 , HIGH);
digitalWrite(28, LOW);
digitalWrite(29, HIGH);
digitalWrite(30, HIGH);
digitalWrite(31 , HIGH);
digitalWrite(32 , LOW);
digitalWrite(33 , LOW);
digitalWrite(34 , HIGH);
digitalWrite(35 , HIGH);
digitalWrite(36 , HIGH);
digitalWrite(37, LOW);
digitalWrite(38, HIGH);
digitalWrite(39, HIGH);
digitalWrite(40 , HIGH);
digitalWrite(41 , LOW);
digitalWrite(42 , LOW);
digitalWrite(43 , HIGH);
digitalWrite(44 , HIGH);
digitalWrite(45 , HIGH);
digitalWrite(46, LOW);
digitalWrite(47, HIGH);
digitalWrite(48, HIGH);
digitalWrite(49 , HIGH);
digitalWrite(50 , LOW);
digitalWrite(51 , LOW);
digitalWrite(52 , HIGH);
digitalWrite(53 , HIGH);

}

void carga_20()
{
```

```
digitalWrite(27 , LOW);
digitalWrite(28, LOW);
digitalWrite(29 , HIGH);
digitalWrite(30, HIGH);
digitalWrite(31 , HIGH);
digitalWrite(32 , LOW);
digitalWrite(33 , LOW);
digitalWrite(34 , HIGH);
digitalWrite(35 , HIGH);
digitalWrite(36 , LOW);
digitalWrite(37, LOW);
digitalWrite(38 , HIGH);
digitalWrite(39, HIGH);
digitalWrite(40 , HIGH);
digitalWrite(41 , LOW);
digitalWrite(42 , LOW);
digitalWrite(43 , HIGH);
digitalWrite(44 , HIGH);
digitalWrite(45 , LOW);
digitalWrite(46, LOW);
digitalWrite(47 , HIGH);
digitalWrite(48, HIGH);
digitalWrite(49 , HIGH);
digitalWrite(50 , LOW);
digitalWrite(51 , LOW);
digitalWrite(52 , HIGH);
digitalWrite(53 , HIGH);
}

void carga_21()
{
digitalWrite(27 , LOW);
digitalWrite(28 , HIGH);
digitalWrite(29, HIGH);
digitalWrite(30, HIGH);
digitalWrite(31 , HIGH);
digitalWrite(32 , HIGH);
digitalWrite(33 , LOW);
digitalWrite(34 , HIGH);
digitalWrite(35 , HIGH);
digitalWrite(36 , LOW);
digitalWrite(37 , HIGH);
digitalWrite(38, HIGH);
digitalWrite(39, HIGH);
digitalWrite(40 , HIGH);
digitalWrite(41 , HIGH);
digitalWrite(42 , LOW);
digitalWrite(43 , HIGH);
digitalWrite(44 , HIGH);
```

```
digitalWrite(45 , LOW);
digitalWrite(46 , HIGH);
digitalWrite(47, HIGH);
digitalWrite(48, HIGH);
digitalWrite(49 , HIGH);
digitalWrite(50 , HIGH);
digitalWrite(51 , LOW);
digitalWrite(52 , HIGH);
digitalWrite(53 , HIGH);
}

void carga_22()
{
    digitalWrite(27 , HIGH);
    digitalWrite(28 , HIGH);
    digitalWrite(29, HIGH);
    digitalWrite(30, HIGH);
    digitalWrite(31 , HIGH);
    digitalWrite(32 , HIGH);
    digitalWrite(33 , LOW);
    digitalWrite(34 , HIGH);
    digitalWrite(35 , HIGH);
    digitalWrite(36 , HIGH);
    digitalWrite(37 , HIGH);
    digitalWrite(38, HIGH);
    digitalWrite(39, HIGH);
    digitalWrite(40 , HIGH);
    digitalWrite(41 , HIGH);
    digitalWrite(42 , LOW);
    digitalWrite(43 , HIGH);
    digitalWrite(44 , HIGH);
    digitalWrite(45 , HIGH);
    digitalWrite(46 , HIGH);
    digitalWrite(47, HIGH);
    digitalWrite(48, HIGH);
    digitalWrite(49 , HIGH);
    digitalWrite(50 , HIGH);
    digitalWrite(51 , LOW);
    digitalWrite(52 , HIGH);
    digitalWrite(53 , HIGH);
}

void semcarga()
{
    digitalWrite(27 , LOW);
    digitalWrite(28, LOW);
    digitalWrite(29 , LOW);
    digitalWrite(30, LOW);
```

```
digitalWrite(31 , LOW);  
digitalWrite(32 , LOW);  
digitalWrite(33 , LOW);  
digitalWrite(34 , LOW);  
digitalWrite(35 , LOW);  
digitalWrite(36 , LOW);  
digitalWrite(37 , LOW);  
digitalWrite(38 , LOW);  
digitalWrite(39 , LOW);  
digitalWrite(40 , LOW);  
digitalWrite(41 , LOW);  
digitalWrite(42 , LOW);  
digitalWrite(43 , LOW);  
digitalWrite(44 , LOW);  
digitalWrite(45 , LOW);  
digitalWrite(46 , LOW);  
digitalWrite(47 , LOW);  
digitalWrite(48 , LOW);  
digitalWrite(49 , LOW);  
digitalWrite(50 , LOW);  
digitalWrite(51 , LOW);  
digitalWrite(52 , LOW);  
digitalWrite(53 , LOW);  
}
```