



Universidade Federal
de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

EMMANUEL AIRES URQUIZA DE CARVALHO

RELATÓRIO DE ESTÁGIO: TRANSFORMADORES CAMPINENSE

Campina Grande, Paraíba
Abril de 2015

EMMANUEL AIRES URQUIZA DE CARVALHO

RELATÓRIO DE ESTÁGIO: TRANSFORMADORES CAMPINENSE

*Relatório de estágio integrado submetido à
Unidade Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.*

Área de Concentração: Processamento de Energia

Orientador:

Professor Tarso Vilela Ferreira, D. Sc.

Campina Grande, Paraíba
Abril de 2015

EMMANUEL AIRES URQUIZA DE CARVALHO

RELATÓRIO DE ESTÁGIO: TRANSFORMADORES CAMPINENSE

*Relatório de estágio integrado submetido à
Unidade Acadêmica de Engenharia Elétrica
da Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.*

Área de Concentração: Processamento de Energia

Aprovado em / /

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Professor Tarso Vilela Ferreira, D. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

Dedico este trabalho aos meus pais. À minha mãe por todos os esforços que fez para que eu pudesse concluir meus estudos. E ao meu pai, que me inculuiu o amor pela ciência e pelo conhecimento, mas que não viveu para ver minha graduação, de modo que, infelizmente, esta dedicatória terá que ser *in memoriam*...

AGRADECIMENTOS

A obtenção do meu título de engenheiro e, talvez de qualquer título, de qualquer pessoa, foi praticamente um trabalho coletivo, tal a quantidade de ajuda que recebi de familiares, amigos, professores e funcionários da UFCG ao longo da minha graduação. Devo, pois, utilizar este espaço para agradecer a todos os que me permitiram chegar aqui.

Aos colegas, professores e funcionários que me ajudaram, gostaria que me perdoassem por não citá-los nominalmente neste espaço. Faço isso porque não confio na minha própria memória e tenho medo de ser ingrato e esquecer algum amigo, já que, felizmente, tenho muitos. Portanto, me dirijo a todos os inestimáveis colegas, funcionários e professores que me ajudaram: foi uma honra.

A minha família certamente merece um agradecimento mais profundo do que o que eu sou capaz de pensar ou escrever neste modesto espaço. Certamente não sou capaz de expressar como gostaria a minha eterna gratidão pelo extraordinário e permanente apoio. O que posso fazer é usar todo o resto da minha vida para honrá-los e transmitir os seus (nossos) valores.

“Aqueles que buscam o sucesso precisam, primeiro, fazer as perguntas certas.”
Aristóteles.

RESUMO

O presente trabalho consiste em um relatório de estágio, o qual elenca as atividades efetuadas entre os dias 11/11/2014 e 25/03/2015 na empresa Transformadores Campinense, pelo estudante Emmanuel Aires Urquiza de Carvalho. Os pontos principais são: o desenvolvimento de ferramentas auxiliares para projetistas (planilhas e software específico), curso de instalação e manutenção de painéis fotovoltaicos no SENAI e o estudo e diagrama de equipamentos adquiridos pela empresa. Outros tópicos menores foram acrescentados, como o projeto do sistema de controle do forno de óleo elétrico, as montagens de alguns ensaios em transformadores e o estudo de diversas ferramentas consideradas promissoras pela administração.

LISTA DE ILUSTRAÇÕES

Figura 1. Layout de um circuito feito pelo Eagle.	13
Figura 2. Layout de um circuito feito pelo Proteus.	13
Figura 3:Diagrama de ligações entre o MAX6675 e um micro controlador.....	14
Figura 4: Programa do circuito que mede e controla a temperatura de um forno de óleo	16
Figura 5: Interface do software para cálculo de consumo por tarifa.....	19
Figura 6: Diagrama esquemático do sistema.	33
Figura 7: Esquema de ligações do ensaio de teste do inversor.	37
Figura 8: Exemplo de implementação em LADDER para a possível lógica da contagem de voltas.	41
Figura 9: Keypad utilizado.	42
Figura 10: imagem do inversor WEG utilizado.....	44

LISTA DE TABELAS

Tabela 1: levantamento de carga do sistema	31
Tabela 2: sentido de rotação do motor em função das entradas STF e STR do inversor.....	37
Tabela 3: comportamento do contador quando motor inicia no sentido horário e muda para o sentido anti-horário na segunda volta.....	38
Tabela 4: comportamento do contador quando motor inicia no sentido anti-horário e muda para o sentido horário na segunda volta.....	39
Tabela 5: tabela da verdade da variável de estados A	39

SUMÁRIO

Agradecimentos.....	v
Resumo.....	vii
Lista de Ilustrações.....	viii
Lista de Tabelas.....	ix
Sumário.....	x
1 Introdução.....	11
2 Desenvolvimento do Estágio.....	12
2.1 Avaliação de softwares para a confecção de PCI.....	12
2.2 Desenvolvimento de um sistema de controle de temperatura de um forno elétrico.....	14
2.3 Medição da quantidade de espiras em um transformador de potencial.....	17
2.4 Ferramentas auxiliares para projetos.....	18
2.4.1 Software para cálculo de consumo.....	18
2.4.2 Planilha Excel.....	20
2.4.3 Gravando Macros no Excel.....	21
2.4.4 Tabelas e dados na planilha excel.....	24
2.5 Curso de instalação e manutenção de painéis fotovoltaicos no SENAI.....	30
2.5.1 Introdução.....	30
2.5.2 Sistemas fotovoltaicos isolados.....	30
2.5.3 Sistemas fotovoltaicos conectados à rede elétrica.....	33
2.5.4 Projeto específico.....	34
2.5.5 Posicionamento dos painéis.....	35
2.6 Testes de equipamentos.....	35
2.6.1 A bobinadeira.....	35
2.6.2 O inversor WEG.....	42
3 Conclusão.....	45
Bibliografia.....	46
APÊNDICE A- Digramas do ensaio de ruptura.....	47
APÊNDICE B- Código em C, para comunicação entre o PIC e a tela de LCD.....	49
APÊNDICE C-Código em C, para o controle da temperatura.....	65
APÊNDICE D-Código em VBA das macros utilizadas na Planilha Excel.....	67
APÊNDICE E-Tabelas utilizadas na Planilha Excel.....	71
APÊNDICE F-Digramas da bobinadeira.....	73

1 INTRODUÇÃO

O presente trabalho consiste em um relatório de estágio, o qual elenca as atividades efetuadas entre os dias 11/11/2014 e 25/03/2015 na empresa Transformadores Campinense, pelo estudante Emmanuel Aires Urquiza de Carvalho.

Parte considerável do estágio foi gasta estudando material sugerido pelos supervisores, assim como, adquirindo competências no uso de diversos softwares para projeto e análise de circuitos.

A empresa Transformadores Campinense está instalada no distrito dos mecânicos em Campina Grande e dispõe de trinta operários, um engenheiro encarregado de projetos e dois secretários, além do estagiário.

Embora atue há mais de vinte anos reformando transformadores a empresa Transformadores Campinense apenas há três anos começou a executar projetos elétricos. Foi neste setor que a maior parte das atividades de estágio foram desempenhadas.

Como a empresa está apenas começando a fazer estes projetos, necessitou criar softwares e ferramentas auxiliares, que pudessem otimiza-los. Parte considerável do tempo do estágio foi utilizada desenvolvendo tais ferramentas ou estudando possíveis ferramentas já existentes.

Embora ainda não faça projetos envolvendo eletrônicos para uso externo, foram desenvolvidos projetos para uso interno à empresa. A empresa também estuda a possibilidade de fazer projetos eletrônicos.

2 DESENVOLVIMENTO DO ESTÁGIO

No decorrer deste Capítulo serão apresentadas as atividades mais significativas desenvolvidas ao longo de todo o período de estágio. Dentre elas, destacam-se:

- i. O estudo de softwares para o desenho de circuitos impressos e de simulação;
- ii. A medição da quantidade de espiras de um transformador de potencial e diagramação do circuito utilizado para o ensaio de ruptura;
- iii. A implementação de ferramentas computacionais auxiliares para projetistas;
- iv. Projeto e simulação do software de controle da temperatura;
- v. Curso de manutenção e instalação de painéis fotovoltaicos no Senai;
- vi. Estudo e diagramação de uma bobinadeira adquirida pela empresa e de um inversor WEG CFW10.

Outras atividades envolvidas foram a construção de uma subestação aérea e uma visita técnica à mineradora Mibra Minérios para efetuar as medições das construções (e, posteriormente, projetar um sistema de proteção contra descargas atmosféricas).

2.1 AVALIAÇÃO DE SOFTWARES PARA A CONFECÇÃO DE PCI

A empresa começou a desenvolver projetos elétricos há cerca de três anos e está examinando as possibilidades de começar projetos eletrônicos. Foi neste setor que houve maior atuação do estagiário. Embora não seja uma necessidade imediata foi pedida uma avaliação de alguns softwares para a confecção de PCI.

Foram testados dois softwares para o desenho de circuitos impressos, são eles: o Proteus e o Eagle. Em ambos existe uma interface de “esquema” em que foi desenhado o circuito na forma de um diagrama convencional de circuitos, com os símbolos apropriados, e uma interface de “layout” de circuito impresso.

Em ambos os softwares, primeiro foi feito esquema em seguida usado um comando para que o layout fosse traçado. No Proteus ainda é possível simular o circuito na parte de esquema. Foi considerado que o Proteus não tão intuitivo quanto o Eagle, sendo esta a sua principal desvantagem.

O Eagle é bem mais simples de se usar, e intuitivo de se aprender. Em bem menos tempo de uso, foi possível desenhar um circuito.

Na Figura 1 e na Figura 2, tem-se exemplos de *layouts* feitos com o Eagle e o Proteus, respectivamente. O circuito é um *dimmer* para controle de tensão baseado no disparo de tiristores.

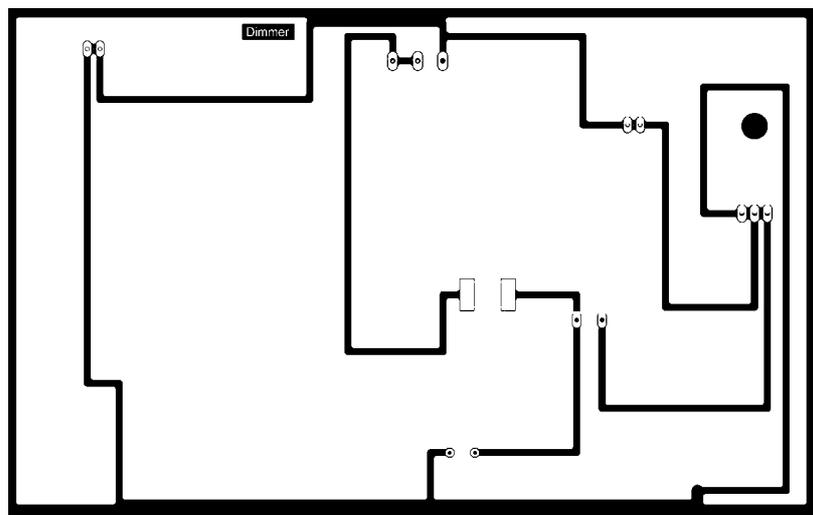


Figura 1. Layout de um circuito feito pelo Eagle.

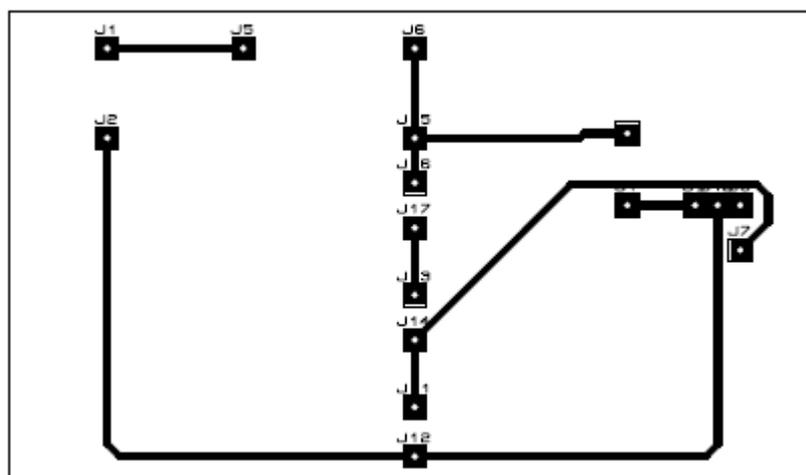


Figura 2. Layout de um circuito feito pelo Proteus.

Nota-se que as trilhas estão demasiadamente espaçadas. Como tratava-se de um projeto de teste, não foram levados em consideração, inicialmente, otimização e estética da PCI.

2.2 DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE DE TEMPERATURA DE UM FORNO ELÉTRICO

Um dos projetos eletrônicos desenvolvidos foi um sistema de controle para a temperatura de óleo de transformador, pois o óleo precisa ser aquecido até cem graus Celsius para que perca umidade. Foi decidida a utilização de microcontroladores para a implementação desse sistema.

O estagiário realizou estudos e testes com o microcontrolador PIC. O estagiário fez, também, um programa básico apenas para fazer o “teste das lâmpadas”. Uma rotina cuja única finalidade era acender cada LED da placa após um breve intervalo de tempo.

Iniciou-se, também, o estudo de possíveis rotinas capazes de ler as informações de um termopar, com a finalidade de medir a temperatura de um tanque de óleo de transformador e, futuramente, utilizar esta informação em uma malha de controle.

Constatou-se que o CI MAX6675 pode servir de conversor digital-analógico entre o termopar e o PIC. O CI recebe a informação de um termopar do tipo “k” e a envia em um sinal de comunicação SPI que pode ser lido pelo microcontrolador, utilizando três pinos. O diagrama elétrico deveria ficar como na Figura 3.

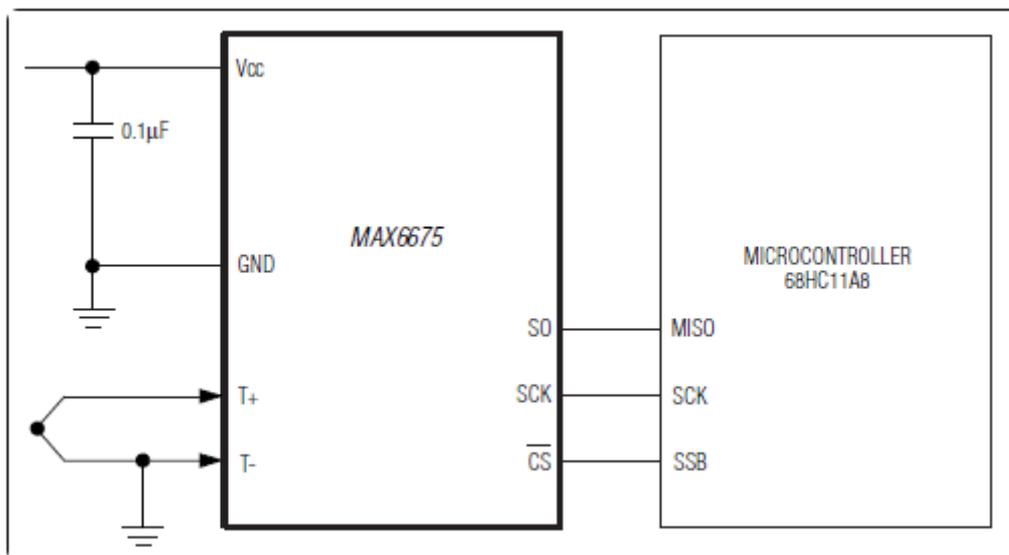


Figura 3: Diagrama de ligações entre o MAX6675 e um microcontrolador.

Fonte: <http://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/MAX6675.html>

Embora o diagrama acima, oriundo do site da companhia e que representa uma aplicação típica do MAX6675, esteja usando um microcontrolador da Motorola é possível fazer o mesmo com um PIC.

Estudou-se, também, o uso de um sensor LM35. Neste caso seriam usadas as entradas analógica-digital do PIC para serem feitas as leituras da saída do sensor.

Prosseguindo a intenção de implementar um sistema de controle automático de temperatura, estudou-se as técnicas usuais para a medição da temperatura por meio de circuitos elétricos, pois o micro controlador precisará da informação elétrica.

O estudo foi concentrado em três métodos de medição de temperatura, e que gere sinais de tensão para que se possa fazer um controle automático: o uso de termopares, o uso dos sensores LM35 e PT100. A grande vantagem do termopar é que ele não se degradaria facilmente no óleo, mas é o método mais caro avaliado. O uso do PT100 pareceu bastante promissor. O sensor consiste em um resistor de platina (Pt, na tabela periódica) que varia conforme a temperatura. Este resistor tem resistência de cem ohms a zero grau Celsius, e varia linearmente.

Seriam necessárias, portanto, poucas medições para calibrá-lo, em seguida seria necessário um simples divisor de tensão para determinar a temperatura. A tensão sobre o sensor, ou sobre o componente não-variável do divisor seria enviada para a porta analógica-digital do PIC e devidamente operada. O problema deste método seria a elaboração de uma forma de proteger o sensor do contato com o óleo, pois os equipamentos revestidos são bastante caros. Uma possibilidade é envolvê-lo em uma tampa metálica e o seu fio estar coberto de material isolante resistente, como borracha.

O mesmo problema foi enfrentado com o LM35, cuja vantagem em relação ao PT100 parece se resumir ao fato de drenar baixas corrente, exigindo uma alimentação de 4 a 20 Volts.

Por fim, optou-se por usar o PT100 devido à sugestão do professor Gutemberg Lira de eletrônica, de que seria possível proteger o sensor com plástico. Esta ideia foi considerada a melhor devido ao custo reduzido.

Utilizando a ferramenta Proteus e o software CCS C Compiler (que é de código aberto), foi desenhado o circuito da Figura 4.

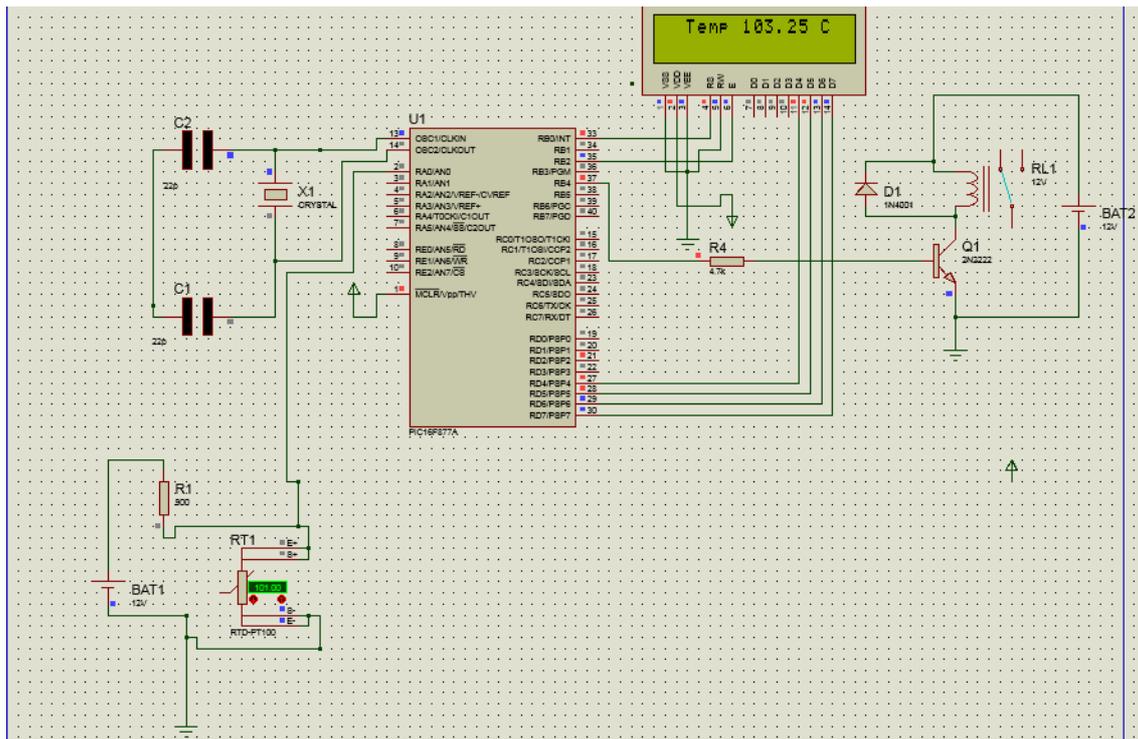


Figura 4: Programa do circuito que mede e controla a temperatura de um forno de óleo

Utilizou-se um resistor variável com a temperatura, PT100. Devido à resolução do PIC e ao fato de que não foram utilizados amplificadores a medição não é precisa demais, entretanto para a aplicação é satisfatória, o estagiário testou por várias horas diversos valores de temperatura e a variação jamais ultrapassou os 4 graus.

O relé mudará de posição quando a temperatura mostrada no display de LCD ultrapassar 105 graus Celsius. Foi utilizado um transistor, um resistor de 4,7 k Ω , um relé de 12 volts e um diodo de roda livre.

Este circuito, embora tenha sido desenhado e simulado com êxito, não chegou a ser montado, de fato ao longo do estágio, pois como não era prioritário e o número de trabalhos urgentes era grande, o projeto apenas foi arquivado para quando for possível fazê-lo. O código, em C, do microcontrolador está no apêndice. O código foi adaptado a partir de outro, de autoria do técnico Lúcio Hélio Santana.

2.3 MEDIÇÃO DA QUANTIDADE DE ESPIRAS EM UM TRANSFORMADOR DE POTENCIAL

A empresa faz reformas de transformadores e recebe produtos muitas vezes danificados ou sem informações adicionais a respeito de sua condição presente, mesmo nos casos em que o transformador aparentemente está em perfeitas condições devem ser feitos ensaios para averiguar se as informações da placa estão corretas. A empresa recebeu um transformador que deveria ser preparado para receber um ensaio de ruptura.

Foi necessário, então, efetuar um ensaio de ruptura em um transformador de potencial, e para tanto era preciso conhecer o número de espiras no transformador.

Embora se dispusesse de um medidor de relação de espiras, conhecido pela sigla em inglês TTR (*Transformer Turns Ratio*), este não serviria para medir as espiras de um Transformador de Potencial (TP), devido à grande quantidade de espiras neste. Foi decidido medir a relação de transformação aplicando uma tensão conhecida no primário, medindo a do secundário e dividindo a primeira pela segunda. Mostrou-se impraticável contar, sem auxílio de equipamentos, as espiras, mesmo as do primário. Então enrolou-se um fio no primário e foi feita uma quantidade conhecida de voltas (setenta voltas). Em seguida utilizou-se o TTR para medir a relação de transformação entre a bobina que feita e o primário do TP. Multiplicado o resultado pelo número de voltas na bobina auxiliar e foi determinado que o primário tem, aproximadamente, 525 voltas.

O estagiário não fez o ensaio final para descobrir a relação de transformação do TP, pois só poderia fazê-lo na presença do supervisor, que não poderia estar presente naquele momento. Ficou decidido que seria feita posteriormente. Entretanto, foi feito pelo supervisor do estágio, mas não na presença do estagiário.

Foi necessário esquematizar o circuito utilizado para o ensaio de ruptura. O circuito está representado no apêndice.

Foi feito, também, outro diagrama, para o entendimento mais fácil do comando do circuito. O circuito também está representado no apêndice.

2.4 FERRAMENTAS AUXILIARES PARA PROJETOS.

Como a empresa faz consultorias e projetos, foi considerado estratégico que se tivesse um software para decidir a melhor tarifa de energia elétrica para os clientes potenciais. Um software certamente economizaria muito tempo e aumentaria a produtividade do setor. A empresa determinou, também, que seria útil ter uma planilha Excel com as características descritas neste tópico, com várias finalidades a serem descritas posteriormente.

2.4.1 SOFTWARE PARA CÁLCULO DE CONSUMO.

Foi escrito um software, utilizando a plataforma Visual Studio 2013, para cálculo do consumo de energia elétrica em cada uma dessas três modalidades tarifárias: branca, verde e azul.

O programa deveria receber as seguintes informações de entrada:

- i. Demanda média e contratada no horário de ponta e fora de ponta;
- ii. Fator de carga de ponta e fora de ponta;
- iii. Quantidade de horas semanais nos horários de ponta e fora de ponta.

A saída deveria ser um gráfico em barras que mostrasse sete barras. Três delas mostrariam o valor total pago por mês com a tarifa verde, a parcela deste valor correspondente à demanda e a parte correspondente ao consumo. Outras três barras farão o mesmo para a tarifa azul. A barra restante será usada para dar o valor total gasto com a tarifa convencional. Como a tarifa convencional não possui parcela correspondente à demanda só é necessária uma barra, que dependerá apenas do consumo. A interface do programa está representada na Figura 5.

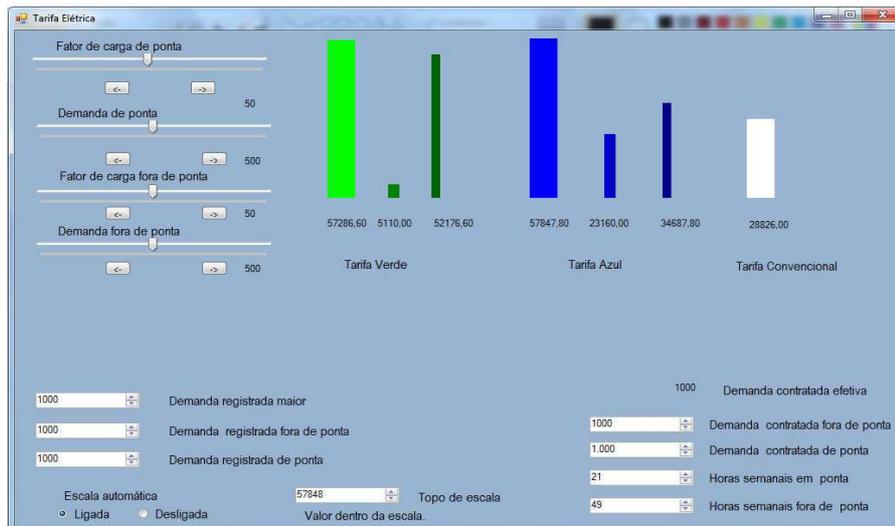


Figura 5: Interface do software para cálculo de consumo por tarifa

O fator de carga está expresso em valor percentual e a demanda em quilowatts. As *trackbars* (as barras com cursores que podem ser arrastados) do lado esquerdo controlam a demanda média e o fator de carga. A demanda registrada é a demanda máxima e será calculada dividindo-se a demanda média pelo fator de carga. Caso os valores das demandas registradas sejam alterados pelas *trackbars* os valores nos mostradores serão atualizados automaticamente e a demanda contratada será igual à demanda registrada. Caso o usuário altere os valores das demandas registradas pelo mostrador a demanda contratada não assumirá o mesmo valor que a demanda registrada. Esta decisão foi tomada porque embora idealmente a demanda registrada não deva ultrapassar a demanda contratada, eventualmente isto acontece e uma multa precisa ser paga (a parcela da multa está considerada na parte correspondente à demanda nos gráficos das tarifas verde e azul).

Foi necessária a criação uma escala ajustável pois a variação do valor a pagar, sobretudo em função do fator de carga varia de forma acentuadamente não-linear, sendo muito maior nos valores mais baixos do fator de carga e variando pouco nos valores maiores, tornando muito difícil perceber variações nos valores mais altos de fator de carga. Foi desenvolvida, então, uma escala que varia de forma automática ou manual. Quando está no modo automático (que é o padrão inicial), o topo da escala corresponde ao valor mais alto dentre os valores numéricos associados às barras. Caso o modo automático esteja desligado pode-se escolher o valor do topo da escala movendo as setas da caixa numérica que indica “topo da escala”. Caso o valor escolhido seja baixo demais,

isto é, caso haja pelo menos uma barra cuja correspondência numérica ultrapasse o topo de escala determinado pelo usuário, aparecerá, abaixo do valor escolhido de topo de escala a frase “valor fora de escala” e as barras não serão alteradas.

Uma característica importante do programa é que, para o cálculo da tarifa foi utilizada apenas uma demanda registrada para encontrar a parcela correspondente à demanda. A demanda registrada utilizada para o cálculo foi a maior das duas (ponta ou fora de ponta). O mesmo acontece para a demanda contratada, na tarifa verde.

No código a tarifa convencional foi chamada de “branca” pois, aparentemente, ela deixará de existir e será substituída pela tarifa branca no futuro. Quando o algoritmo foi iniciado ainda não se sabia se ele seria feito com base na tarifa convencional ou com base na tarifa branca.

Um fato importante é que, tanto na tarifa azul como na verde, o cálculo da parcela de demanda é feito com base nos valores das demandas registrada e contratada. Se a contratada exceder a registrada, utiliza-se a contratada. Caso a registrada exceda a contratada usa-se a registrada. Além disso, se a registrada exceder a contratada em mais do que cinco por cento, paga-se uma multa.

Na demanda azul o procedimento é similar, porém faz-se o teste com as demandas registradas de ponta e fora de ponta e com as demandas contratadas de ponta e fora de ponta. Por exemplo: caso a demanda registrada de ponta seja maior que a demanda contratada de ponta o valor usado no cálculo da parcela de demanda será o da demanda de ponta, similar e independentemente ao que ocorrerá entre as demandas registrada e contratada fora de ponta.

2.4.2 PLANILHA EXCEL.

O objetivo do projeto é o desenvolvimento de uma planilha Excel que receba como dados de entrada: demanda registradas de ponta e fora de ponta, demanda reativa, consumo de ponta, demanda contratada de ponta, demanda contratada fora de ponta, consumo fora de ponta, consumo intermediário, horas ativas de ponta, horas ativas fora de ponta e horas ativas intermediárias.

Deve produzir gráficos comparados dos custos para cada tipo de tarifação. Também deve produzir gráficos comparados da demanda registrada com a demanda média para situações de ponta e fora de ponta. Por fim, deve produzir gráficos do consumo mensal para situações de ponta e fora de ponta.

A planilha deve possuir ícones para as opções de gráfico e, ao clicar-se neles, será escolhido o gráfico que aparecerá. Para isto serão usadas macros do Excel. Uma macro (abreviação de “macro-instrução”) é definida como “ uma regra ou padrão que especifica como uma certa sequência de entrada (frequentemente uma sequência de caracteres) deve ser mapeada para uma substituição de sequência de saída (também frequentemente uma sequência de caracteres) de acordo com um procedimento definido”.

2.4.3 GRAVANDO MACROS NO EXCEL.

Para gravar-se uma macro deve-se primeiro habilitar a aba de “desenvolvedor” do Excel, que também será utilizada para implementar ícones de controle de formulário, como os que serão usados para escolher os gráficos. Para isto clica-se em “Arquivo”, depois em “Opções”, “Personalizar faixa de opções”, marca-se a opção onde está escrito “desenvolvedor” e clica-se em “OK”, para habilitar a aba. Em algumas versões do Excel esta aba já vem habilitada, embora em outras não.

Cria-se então um gráfico de linhas qualquer aleatório (posteriormente será explicado porque o gráfico pode ser aleatório) e inicia-se a gravação da macro. Para isto, clica-se na aba “exibição”, depois em “Macros” e depois em “gravar macro”. Aparecerá uma opção para nomear a macro e escolher uma tecla de atalho para ele, como o objetivo é apenas executar a macro através de ícones, então não serão estabelecidas teclas de atalho. A partir desse ponto todas as ações serão gravadas na macro. Modifica-se os dados do gráfico para outro de interesse. Supondo que se esteja fazendo a macro “custos” cujos dados utilizados são os valores totais a pagar de cada tarifa para certa demanda (em um tópico posterior serão descritas cada tabela usada e suas propriedades). Clica-se na tabela com o botão direito do mouse, clica-se em “selecionar dados”, no espaço onde há “entradas de legenda” clica-se em cada opção e exclui-se cada uma delas, então clica-se adicionar e depois disso clica-se onde está escrito “nome da série”, seleciona-se com o teclado em seguida clica-se onde está o nome do gráfico, logo após clica-se na opção “valores da série” até aparecer o cursor do teclado e, em seguida, seleciona-se a coluna de valores correspondente. Repete-se este procedimento para cada curva que deseja-se que apareça no gráfico. Por fim clica-se em “Macros” e na opção “Parar Gravação”.

Tem-se agora uma macro pronta. Entretanto é necessário analisar mais detalhadamente o que esta macro faz. Ela transforma o gráfico inicial (que foi estabelecido como aleatório) no seu gráfico final. E executa cada movimento feito por

pela barra de rolagem. Presumivelmente obter-se-ia resultados diferentes para pontos de partida diferentes. Pois a macro executaria os mesmos passos para qualquer que fosse o gráfico inicial, conduzindo a resultados diferentes ou erros. A próxima etapa é editar a macro de modo à sempre chegar ao resultado final desejado, independente de qual seja o gráfico inicial.

Para saber o que deve-se mudar precisa-se saber o que será constante e o que será variável de um gráfico para outro. Constante serão o tipo de gráfico, que será sempre de linhas e os rótulos do eixo horizontal que será sempre de um a doze, correspondendo aos meses do ano. A quantidade de curvas em cada gráfico e os dados de entrada serão variáveis o que é um problema, que precisa ser resolvido. Uma macro que foi criada transformando um gráfico de duas curvas em um de três curvas, não conseguirá partir de um gráfico de uma curva. Pois para fazer a partir de duas, é necessário deletar as duas curvas, e quando ao partir de apenas uma o compilador não acha o objeto a ser deletado. É necessário modificar a macro para que ela, antes de colocar as entradas de legenda desejadas, delete todas as entradas de legendas, independentemente da quantidade.

Clica-se mais uma vez na opção “Macros”, em seguida em “exibir macros”, escolhe-se a macro criado e clica-se na opção “editar”. Será mostrado em uma janela o código em VBA que gera a macro. Se a gravação da macro foi feita de acordo com o procedimento supracitado haverá no código linhas parecidas onde estão escrito coisas similares a “*ActiveChart.SeriesCollection(1).Delete*”, onde o “um” entre parênteses identifica como o primeiro elemento da lista de “entradas de legenda”. Se o gráfico inicial tinha uma curva, haverá uma linha igual a esta, se forem duas curvas, duas linhas iguais a esta e assim sucessivamente. Deve-se apagar estas linhas e construir um comando que determine que a macro deva excluir todas as “entradas de legenda”, independente de quantas tenham e não mais do que isso. Para isto deve-se uma sub-rotina. O nome da sub-rotina é “*temp*”. Apaga-se todas as linhas de código onde esteja escrito “*ActiveChart.SeriesCollection(1).Delete*” e coloca-se “*Call temp*”, no lugar. Após o final da macro, escreve-se a sub-rotina. O código da sub-rotina está escrito abaixo:

```
Public Sub temp()
    ActiveSheet.ChartObjects("Gráfico 2").Activate
    For Each s In ActiveChart.SeriesCollection
        s.Delete
    Next s
```

End Sub

Na sub-rotina está escrito “Gráfico 2” pois este é o nome do gráfico usado. É oportuno lembrar que, uma vez que todas as modificações foram feitas sobre um mesmo gráfico, o nome do gráfico não pode ser mudado, pois é utilizado pelo programa.

Outro problema, mais simples de ser corrigido, é o movimento que é feito pelas barras de rolagem vertical e horizontal do Excel. Durante a gravação da macro, normalmente arrasta-se o mouse até as colunas onde estão os dados e todo este movimento é gravado na macro. O movimento que o usuário faz é relativamente aleatório, mas ele gerará um código similar ao descrito abaixo:

```
ActiveWindow.ScrollColumn = 13
ActiveWindow.ScrollColumn = 12
ActiveWindow.ScrollColumn = 11
ActiveWindow.ScrollColumn = 10
ActiveWindow.ScrollColumn = 9
ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollRow = 19
ActiveWindow.ScrollRow = 18
ActiveWindow.ScrollRow = 17
ActiveWindow.ScrollRow = 16
ActiveWindow.ScrollRow = 15
ActiveWindow.ScrollRow = 14
ActiveWindow.ScrollRow = 13
```

A quantidade de linhas e os valores associados vai depender do tipo de movimento que o usuário fez, mas o parâmetro *ActiveWindow.ScrollColumn*, controla o movimento vertical e o parâmetro *ActiveWindow.ScrollRow*, controla o movimento horizontal. Como deseja-se eliminar movimento de qualquer tipo, todas estas linhas foram deletadas.

Outro tipo de código indesejado que ficará guardado na gravação, e também dará a ilusão de aparente movimento, é o ato de clicar em uma célula qualquer antes de terminar a gravação da macro. Caso o programador clique na célula “C25”, por exemplo, aparecerá no código a linha “*Range("C25").Select*”, e quando a macro for executada isso significará que esta célula vai receber o cursor do mouse, podendo deslocar a barra de rolagem. Como nesta aplicação estes movimentos são indesejáveis, apagam-se, no código, as linhas onde esta função é chamada.

Para chamar a macro, clica-se na aba “desenvolvedor” e logo após em “inserir”, escolhe-se um “botão (controle de formulário)”, e então escolhe-se macro a ser executada quando o ícone for clicado.

Este procedimento foi repetido seis vezes, para os seis ícones. Os gráficos mostrados são (as macros correspondentes estão no apêndice no final do relatório):

- Custos: três curvas (Valor total a pagar com a tarifa azul, com a tarifa verde e com a tarifa branca).
- Demanda ponta: duas (demanda registrada e demanda média, ambas de ponta).
- Demanda f.ponta: duas (demanda registrada e demanda média, ambas fora de ponta).
- Demanda reativa: uma (apenas a demanda reativa).
- Consumo ponta: uma (apenas o consumo em ponta).
- Consumo f.ponta: uma (apenas o consumo em fora de ponta).

As tabelas de onde foram obtidos os valores das curvas supracitadas serão explicados no próximo tópico.

2.4.4 TABELAS E DADOS NA PLANILHA EXCEL.

Conforme já foi dito na introdução os parâmetros de entrada são: demanda registradas de ponta e fora de ponta, demanda reativa, consumo de ponta, demanda contratada de ponta, demanda contratada fora de ponta, consumo fora de ponta, consumo intermediário, horas ativas semanais média de ponta, horas ativas semanais média fora de

ponta e horas ativas semanais médias intermediárias. Cada um desses dados indica o nome de um conjunto de doze elementos, cada um deles correspondendo ao valor do parâmetro em um determinado mês. Utilizando estes dados de entrada.

Foram traçadas quatro tabelas: uma principal com todas as entradas e alguns parâmetros adicionais, e três para cada tipo de tarifa, onde cada componente são resultados de operações com dados da tabela principal. A seguir, cada tabela será analisada.

2.4.4.1. Tabela Principal

As colunas cujos títulos estão em vermelho correspondem a saídas, ou seja, dados gerados a partir de operações com os dados de entrada, que estão nas colunas cujos títulos são pretos. Por ser uma tabela muito grande foi necessário dividi-la em duas para se adequar a este documento. A tabela principal está representada nos apêndices.

A seguir serão explicadas as operações realizadas para cada dado de saída.

- 1) *Dem.Reg.Maior*: recebe o valor da maior demanda registrada entre demanda registrada de ponta e a fora de ponta.
- 2) *Ultrap.*: recebe o valor de “*Dem.Reg.Maior*” menos “*demFP.contratada*”. Ou seja recebe o valor da maior demanda registrada entre a demanda de ponta e a fora de ponta e o subtrai da demanda fora de ponta contratada.
- 3) *Dmed.Fponta*: recebe o consumo fora de ponta dividido pelas horas semanais médias de ponta vezes quatro (as horas que estão multiplicadas por quatro, não o total).
- 4) *Dmed.Ponta*: mesmo que o item anterior mas para o consumo de ponta.
- 5) *Dmed.Intermediário*: mesmo que o item anterior mas para o consumo intermediário.
- 6) *Ultrap.Ponta*: demanda registrada de ponta menos demanda contratada de ponta.
- 7) *Ultrap.FPonta*: demanda registrada fora de ponta menos demanda contratada fora de ponta.

2.4.4.2. Tabela da tarifa verde

A tabela correspondente à tarifa verde está nos apêndices.

Todas as colunas desta tabela são saídas, exceto a coluna “mês”, que sempre deve permanecer constante, com os números de um a doze. Cada coluna de saída será descrita a seguir:

- 1) *P.Dem*: parcela do valor pago correspondente a demanda. A operação efetuada é expressa pelas equações 1 e 2, de acordo com a condição correspondente. Se a ultrapassagem de ponta for maior que zero

$$P.Dem.Ponta = 5,11 \times (Dem.Reg.Maior). \quad (1)$$

Caso contrário

$$P.Dem.Ponta = 18,05 \times (Dem.reg.ponta - ultrap..) \quad (2)$$

- 2) *P.Cons.*: parcela do valor pago correspondente ao consumo. *P.cons*, definida pela equação 3.

$$P.Cons = (consumo.ponta) \times 0,76 + (consumof.ponta) \times 0,2067 \quad (3)$$

P.multa: parcela da tarifa correspondente a multa caso a demanda registrada ultrapasse em cinco por cento a demanda contratada. A operação efetuada é a descrita pelas equações 4 e 5

$$P.multa = 5,11 \times 2(dem.Reg.Maior - DemFPcontratada)) \quad (4)$$

Caso

contrário:

$$P.multa = 0. \quad (5)$$

- 3) *Total Verde*: parcela correspondente a soma da todas as parcelas anteriores e que indica o valor total a ser pago, caso o usuário opte pela tarifa verde.

2.4.4.3. Tarifa Azul

A tabela correspondente à tarifa azul está nos apêndices.

Todas as colunas desta tabela são saídas, exceto a coluna “mês”, que sempre deve permanecer constante, com os números de um a doze. Cada coluna de saída será descrita a seguir:

- 1) *P.Dem.Ponta*: parcela correspondente à demanda de ponta. A operação efetuada obedece às equações 6 e 7, dependendo das respectivas condições.

Se a ultrapassagem de ponta for maior que zero:

$$P.Dem.Ponta = 18,05 \times (Dem.reg.ponta) \quad (6)$$

Caso contrário:

$$P.Dem.Ponta = 18,05 \times (Dem.reg.ponta - ultrap.ponta) \quad (7)$$

- 2) *P.Dem.fponta*: parcela correspondente à demanda fora de ponta. A operação efetuada obedece às equações 8 e 9, dependendo das respectivas condições.

Se a ultrapassagem fora de ponta for maior do que zero:

$$P.Dem.fponta = 5,11 \times Dem.reg.fponta \quad (8)$$

Caso contrário:

$$P.Dem.fponta = 5,11 \times (Dem.reg.fponta - ultrap.fponta) \quad (9)$$

- 3) *P.Consumo ponta*: parcela correspondente ao consumo em ponta. A operação efetuada corresponde à equação 10.

$$P.Consumoponta = consumoponta \times 0,3436 \quad (20)$$

- 4) *P.Consumo fp*: parcela correspondente ao consumo fora ponta. A operação efetuada corresponde à equação 11.

$$P.Consumo .fp = consumof .ponta \times 0,2067 \quad (31)$$

P. Multa ponta: parcela correspondente a multa caso a demanda registrada de ponta exceda em mais de 5% demanda contratada de ponta. A operação efetuada obedece às equações 12 e 13, dependendo das respectivas condições. Caso a demanda registrada de ponta exceda em mais de 5% demanda contratada de ponta:

$$Parcela.Multiponta = (dem.Re g.Ponta - DemPcontratada) \times 2 \times 18,05 \quad (42)$$

Caso contrário:

$$Parcela.Multiponta = 0 \quad (53)$$

- 5) *P. Multa f.ponta*: parcela correspondente a multa caso a demanda registrada fora de ponta exceda em mais de 5% demanda contratada fora de ponta. A operação efetuada obedece às equações 14 e 15. Caso a demanda registrada fora de ponta exceda em de 5% demanda contratada fora de ponta:

$$P.Multa.f.ponta = (dem.Re g.fp - DemFPcontratada) \times 2 \times 5,11 \quad (14)$$

Caso contrário:

$$P.Multa.f.ponta = 0. \quad (65)$$

6) *Total Azul*: parcela que representa o custo total da energia com esta tarifa. Corresponde à soma de todas as parcelas anteriores.

- Tarifa Branca

A tabela correspondente à tarifa branca está nos apêndices.

Todas as colunas desta tabela são saídas, exceto a coluna “mês”, que sempre deve permanecer constante, com os números de um a doze. Cada coluna de saída será descrita a seguir:

1) *P.Consumo .ponta*: parcela correspondente ao consumo em ponta. A operação efetuada está expressa na equação 16.

$$P.Consumo .ponta = consumo_{ponta} \times 0,2059 \quad (76)$$

2) *P.Consumo f.ponta* parcela correspondente ao consumo fora ponta. A operação efetuada está expressa na equação 17.

$$P.Consumo f.ponta = consumo_{f.ponta} \times 0,10295 \quad (87)$$

3) *P.Consumo intermediário*: parcela correspondente ao consumo intermediário. A operação efetuada está expressa pela equação 18.

$$P.Consumo.int ermediario = consumo.int ermediario \times 0,154425 \quad (98)$$

4) *Total Branca*: parcela correspondente à soma de todas as parcelas anteriores.

Os códigos em VBA dos macros utilizados está no apêndice.

2.5 CURSO DE INSTALAÇÃO E MANUTENÇÃO DE PAINÉIS

FOTOVOLTAICOS NO SENAI

A empresa planeja diversificar suas atividades em projetos elétricos. Como o Senai disponibilizou cursos gratuitos de manutenção e instalação de painéis fotovoltaicos, foi determinado que o estagiário deveria fazer este curso e escrever um resumo do conteúdo.

2.5.1 INTRODUÇÃO.

Durante duas semanas do estágio foi realizado um curso de manutenção e instalação de painéis fotovoltaicos do Senai. Deveria ser produzido um relatório sobre o conteúdo do curso. A parte inicial do curso foi constituída apenas de conceitos elementares de eletricidade e segurança do trabalho. A parte específica foi focada no dimensionamento de sistemas isolados e conectados à rede.

2.5.2 SISTEMAS FOTOVOLTAICOS ISOLADOS.

São chamados sistemas fotovoltaicos isolados aqueles que não se conectam com a rede elétrica (*off grid*). Os sistemas isolados podem ou não possuir bancos de baterias. Caso não possuam, não serão capazes de suprir a carga caso a geração diminua em decorrência das condições climáticas. Caso existam baterias será necessário, também, um controlador para fazer com que o sistema exija potência da bateria ou dos painéis no momento adequado. As baterias devem ser sempre do tipo estacionárias. Nunca deve-se usar baterias de automóvel ou outro tipo de bateria para esta aplicação. Todos os painéis devem possuir as mesmas especificações entre si, pois o sistema seria limitado pelo painel de menor potência.

Abaixo segue um exemplo do dimensionamento de um sistema isolado, na forma como este é usualmente especificado.

O sistema terá o seguinte levantamento representado pela Tabela 1.

Tabela 1: levantamento de carga do sistema

Item	Tempo de uso(h)	Potência(W)	Energia gasta em um dia(Wh/dia)
Lâmpada 1	4	25	100
Lâmpada 2	5	25	125
Televisão	5	40	200
Geladeira	10	80	800
Rádio	6	100	600

E as seguintes especificações:

- Autonomia de dois dias, isto é, caso cesse a geração as baterias devem manter o sistema funcionando por dois dias
- Profundidade de descarga de 40%, isto é a bateria tem sempre 60% de sua carga. A maioria das baterias precisam ser operadas nesta profundidade de descarga sob pena de severa diminuição de sua vida útil.
- Considerando média de insolação do local de 4 horas de sol por dia (site “sundata”).

Primeiro determina-se a energia total consumida por dia: $E=100+125+200+800+600=1825\text{Wh}$. A seguir dimensiona-se o banco de baterias.

O banco deve suprir a carga por dois dias e se descarregar em apenas 40%, logo:

$$Energia\ do\ banco = \frac{Energia_total \times dias_de_autonomia}{profundidade\ de\ descarga} = \frac{1825 \times 2}{0,4} = 9125\ Wh$$

Esta é a energia que o banco de baterias precisa ser capaz de fornecer. A capacidade das baterias é dada em Ah (ampère-hora), se foram escolhidas baterias de doze volts, tem-se que a capacidade do banco de baterias será:

$$Capacidade\ do\ banco = \frac{Energia\ do\ banco}{tensão\ de\ cada\ bateria} = \frac{9125}{12} = 760,42\ Ah$$

Se escolher-se baterias de 220Ah, tem-se:

$$Quantidade = \frac{capacidade\ do\ banco}{capacidade\ de\ cada\ bateria} = \frac{760,42}{220} = 3,456 \approx 4\ baterias$$

Agora, dimensiona-se os módulos fotovoltaicos.

Primeiro precisa-se achar a potência fotovoltaica:

$$\text{Potência fotovoltaica} = \frac{\text{Energia}_{total}}{\text{horas de insolação diárias}} = \frac{1825}{4} = 456,25 \text{ W}$$

Utilizou-se painéis de 160 W, logo:

$$\text{Quantidade de painéis} = \frac{\text{potência fotovoltaica}}{\text{potência de cada painel}} = \frac{456,25}{160} = 2,85 \approx 3 \text{ painéis}$$

Agora dimensiona-se o controlador.

- A tensão do controlador deve ser a mesma das baterias, portanto, 12V
- A corrente do controlador deve ser limitada pela soma das correntes de pico proveniente dos painéis. A corrente de pico dos painéis é especificada no painel. Existem duas formas de dimensionamento, uma é utilizando a corrente de curto-circuito do painel, outra é usando a corrente de máxima potência e multiplicando o resultado por um fator de 1,2 de segurança. Supondo que será feito da segunda maneira, utilizando uma corrente de máxima potência de 13,3A. Os painéis estão em paralelo, logo as correntes irão se somar. Então:

$$I_{controlador} = 13,3 \times 3 \times 1,2 = 47,88$$

O controlador deve suportar ao menos isto.

Uma informação importante acerca de controladores é que alguns possuem “seguidor de máxima potência” (não confundir com “seguidor solar”), chamado de MPPT. Os cálculos feitos até aqui consideram que o controlador possui este sistema. A única mudança caso não seja utilizado um controlador MPPT será no cálculo da potência fotovoltaica, que deverá receber um acréscimo de 30%, ou seja:

$$\begin{aligned} \text{Potência fotovoltaica} &= \frac{\text{Energia}_{total}}{\text{horas de insolação diárias}} + \frac{\text{Energia}_{total}}{\text{horas de insolação diárias}} * 0,3 \\ &= 456,25 \times 0,3 + 456,25 = 593,125 \end{aligned}$$

O que aumentaria a quantidade de painéis.

O diagrama esquemático do sistema está representado na Figura 6.

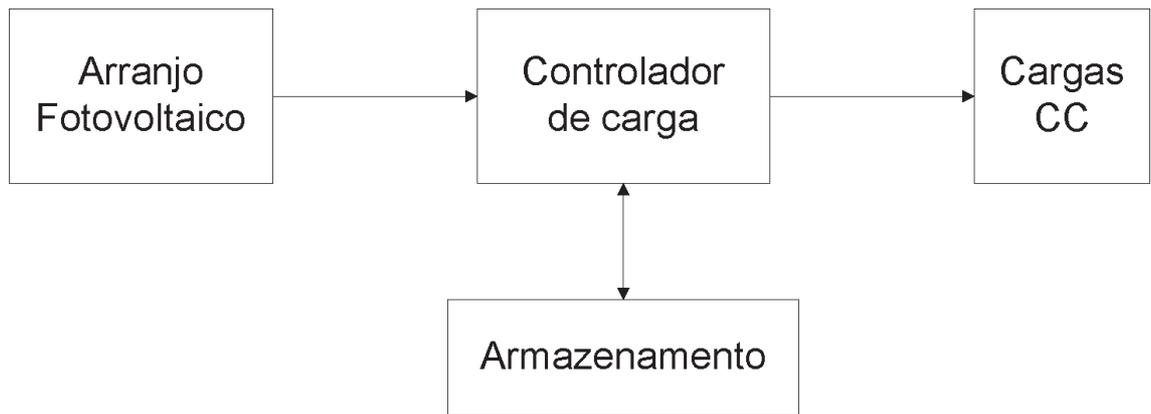


Figura 6: Diagrama esquemático do sistema.

O controlador possui entradas específicas para os painéis, as baterias e a carga. Colocar componentes fora das entradas especificadas pode danificar o controlador.

Caso o sistema abastecesse uma carga que necessitasse de tensão alternada seria necessário colocar um inversor entre o controlador e a carga. E o inversor seria dimensionado para 1,2 vez a potência da carga.

2.5.3 SISTEMAS FOTOVOLTAICOS CONECTADOS À REDE ELÉTRICA

O sistema conectado à rede usará obrigatoriamente um inversor do tipo *tie-grid*, um sistema de painéis e um contador bidirecional. É necessária autorização da concessionária para ligar o inversor à rede e para trocar o medidor. O princípio básico dos sistemas conectados à rede é que a energia produzida será enviada para a rede, acarretando diminuição da conta de energia. Caso a produção torne-se maior que o consumo, o excedente servirá de crédito por um prazo fixo (provavelmente será de 60 dias), caso o crédito não seja utilizado até o fim desse prazo, será perdido.

Para fins de dimensionamento, basta saber o consumo de energia e o quanto pretende-se diminuir este consumo.

Exemplo:

Para um consumo mensal de 120kWh, deseja-se dimensionar um sistema para ter um abatimento de 70%. Ou seja:

$$\text{Geração mensal} = 120 \times 0,7 = 84 \text{ kWh}$$

Supondo 5,09 horas de sol pleno por dia. O site “sundata” fornece esta informação em 5,09kWh/metro_quadrado-dia, mas a potência máxima dos painéis é dada para um fornecimento de energia de 1kW/metro_quadrado, logo basta dividir

$$\text{horas de sol pleno} = \frac{5,09kWh}{\frac{m^2 - dia}{1kW/m^2}} = 5,09 \text{ horas de sol pleno}$$

Escolhe-se um painel de 240W. A energia gerada por este painel em um mês será:

$$\text{Energia} = 240 \times 5,09 \times 30 = 36,6kWh/mês$$

Logo a quantidade de painéis necessária para conseguir o abatimento necessário é de:

$$\begin{aligned} \text{Quantidade de painéis} &= \frac{\text{Energia que precisa ser gerada em um mês}}{\text{Energia mensal gerada por um painel}} = \frac{84}{36,6} \\ &= 2,3 \approx 3 \text{ painéis} \end{aligned}$$

Para o dimensionamento do inversor é necessário obedecer à seguinte condição:

$$\text{Fator de dimensionamento do inversor (FDI)} = \frac{\text{Potência do inversor}}{\text{Potência da geração}}$$

$$0.9 < FDI < 1,1 \text{ (o ideal é que seja 1)}$$

O inversor deve ser escolhido de modo que esta condição seja satisfeita. Outra informação importante é que estes inversores possuem limites inferiores e superiores de tensão de entrada. Se a tensão de entrada ficar abaixo do limite inferior o inversor não funciona, caso seja superior, o inversor será danificado.

2.5.4 PROJETO ESPECÍFICO

Durante o curso de painéis fotovoltaicos o estagiário recebeu um convite para fazer um projeto com um cliente em São José do Egito. O objetivo é fornecer energia elétrica, através de painéis fotovoltaicos, a um sistema que bombeie água para o telhado, onde ela é aquecida pela luz do sol. O cliente havia feito sozinho as conexões, danificando o inversor, e gostaria de saber como minimizar o prejuízo. Avaliou-se que o problema central é que sistemas isolados com bancos de baterias só são viáveis em locais remotos sem acesso a rede elétrica. O custo de trocar as baterias acaba sendo maior do que a compra de energia pela rede. O cliente usou baterias de automóvel, que não são adequadas para estes propósitos.

Para minimizar o prejuízo inicialmente supôs-se que a solução ideal era apenas fazer as conexões adequadas e, quando as baterias acabassem, recomendar a ele vender o equipamento. Entretanto, descobriu-se posteriormente, que as baterias ultrapassaram a data de validade. Não se pode utilizar baterias vencidas em projetos elétricos, pois poderiam colocar em risco a vida humana no local.

2.5.5 POSICIONAMENTO DOS PAINÉIS

Os painéis fotovoltaicos precisam ser posicionados de modo a otimizar a absorção de energia solar. Para isto deve-se escolher adequadamente a inclinação do painel em relação a horizontal e a direção em que ele está posicionado.

A inclinação deve ser igual à latitude do local onde o painel está instalado, desde que esta latitude seja superior a 10 graus. Caso seja inferior, a inclinação do painel com a horizontal deve ser de dez graus, pois o painel não pode ficar inclinado menos do que isso, devido ao efeito danoso do acúmulo de poeira e água da chuva.

Caso o painel esteja posicionado em algum lugar do hemisfério sul ele deve apontar para o norte, caso esteja no hemisfério norte, deve apontar para o sul. Para isto usou-se uma bússola e um mapa que determine a declinação magnética na região. As bússolas não apontam para o “norte geográfico”, que é a informação de interesse, e sim para o “norte magnético”. Mas, pela utilização de mapas que dizem a declinação magnética em uma região, pode-se posicionar corretamente o painel.

2.6 TESTES DE EQUIPAMENTOS

É praxe na empresa que cada equipamento adquirido seja estudado, tenha um relatório que explique seu funcionamento e, caso seja possível, tenha suas conexões desenhadas. Este procedimento foi efetuado com uma bobinadeira e um inversor WEG CFW10.

2.6.1 A BOBINADEIRA

Este relatório apresenta os resultados do estudo da bobinadeira adquirida pela Transformadores Campinense. Deveria-se descrever seu funcionamento e fazer um

diagrama elétrico dela. Além disso, a bobinadeira quebrou devido a uma ligação incorreta (deveria ter sido ligada com três fases e o neutro e foi ligada sem o neutro). Parte do trabalho consiste em identificar a causa do problema.

Como não era desejável, pelo menos inicialmente, desmontar a máquina e os fios possuíam marcações, deduziu-se que estas marcações identificavam os fios e utilizou-se isto como referência para não ser necessário desmontar o equipamento todo. Mesmo em casos aparentes em que dois fios tinham a mesma marcação, avaliou-se que estes fios ligavam os mesmos componentes (testes de continuidade permitiram isto).

Inicialmente foi feito o diagrama elétrico da bobinadeira em AutoCAD. Para facilitar o estudo do funcionamento, fez-se diversos diagramas mostrando as ligações individuais dos componentes do quadro de controle da bobinadeira. Sendo estes componentes: o inversor, o CLP, o disjuntor e o contactor. O motor está conectado por uma correia de borracha ao eixo que é efetivamente usado para o bobinamento. Conectados a este eixo estão dois sensores. Cada sensor, provavelmente, emite um pulso de tensão quando o parafuso, atrelado ao eixo, passa pelo sensor. O sistema conta, também, com pedais, que são usados para controlar o sentido de rotação e a velocidade do giro. Os diagramas elétricos produzidos estão nos apêndices.

O inversor utilizado foi produzido pela empresa Mitsubishi e sua especificação é FR-A044-1.5K-UL. As entradas utilizadas e suas respectivas funções foram:

- SD: comum do sistema de controle.
- 2,5 e 4. São os terminais nos quais deve-se conectar o potenciômetro utilizado para controlar a frequência do sinal de saída.
- RL/OH. Utilizado para desligar o inversor, normalmente em caso de emergência, quando RL/OH recebe um sinal, de um relé, por exemplo, cujo comum esteja em SD, o inversor parará de operar (*contact break*). Como este controle, no caso da bobinadeira, é feito pelo CLP, conectou-se o pino RL/OH à saída do clp em questão e o pino SD ao pino comum referente à saída do clp utilizada.
- STF e STR: pinos usados para controle do sentido da rotação. Caso utilize-se um relé, por exemplo, deve-se ligar uma ponta do relé em STF e a outra em SD. O mesmo para STR, utilizando outro relé, obviamente. A tabela que descreve o controle do sentido de rotação do motor é a Tabela 2.

Tabela 2: sentido de rotação do motor em função das entradas STF e STR do inversor.

STF	STR	Sentido de rotação
0	0	Parado
0	1	Inverso
1	0	Direto
1	1	Parado

- P e PR: terminais do resistor de frenagem.
- R, S e T: entradas de alimentação trifásica.
- U, V e W: saídas trifásicas.

Para testar o funcionamento do inversor, inicialmente foram feitas as conexões mostradas na Figura 7.

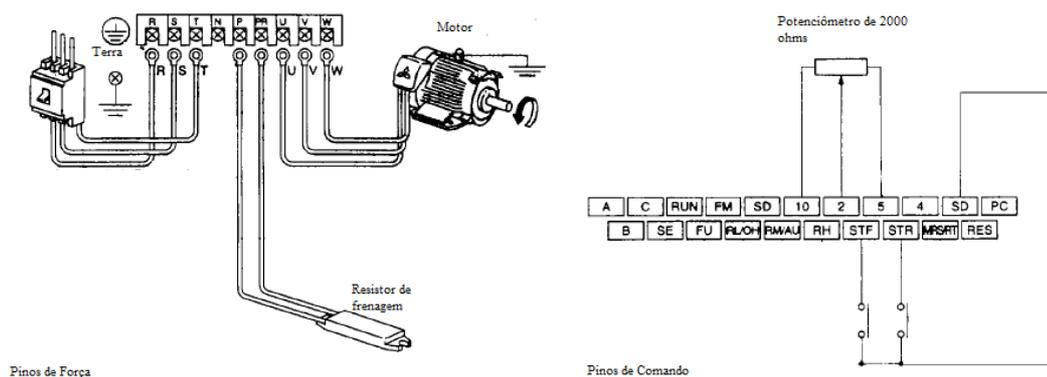


Figura 7: Esquema de ligações do ensaio de teste do inversor.

Em seguida ligou-se a chave da alimentação, o motor estava parado pois os disjuntores STF e STR estavam ambos fechados, então abriu-se o disjuntor do pino STF e o motor começou a girar, variou-se o potenciômetro e o motor mudou de velocidade.

Depois fechou-se o disjuntor do pino STF e abriu-se o do pino STR e o motor passou a girar no sentido inverso. Novamente variou-se o potenciômetro e verificou-se a mudança na velocidade de rotação. Em seguida abriu-se o disjuntor do pino STF sem fechar o do pino STR, e o motor parou de girar. Isto está em acordo com o funcionamento esperado do inversor. O motor utilizado não foi o da bobinadeira mas sim um motor de indução que a empresa já possuía. Verificou-se desta maneira que o inversor estava funcional.

No sistema da bobinadeira os pinos STF, STR e RL/OH estão conectados ao CLP, ou seja, o CLP controla o sentido do giro e pode frear o motor. As entradas do potenciômetro estão conectadas diretamente a um pedal, que controla a velocidade de giro, portanto. A entrada SD está devidamente conectada ao comum das saídas usadas pelo clp.

O modelo de CLP utilizado foi o Fxos-10MR-ES da Mitsubishi.

Analisa-se a seguir o possível uso de cada uma das entradas e saídas usadas do CLP.

- As entradas X0 e X1 estão conectadas a sensores localizados no eixo de rotação da bobinadeira, e servem, presumivelmente, para contar a quantidade de voltas. São necessários dois sensores pois esta é a única forma de inferir o sentido de rotação, ou seja, se o CLP irá efetuar a soma ou subtração da quantidade de voltas. Para se permitir o entendimento da lógica implementada foi feita uma suposição inicial. Supondo que sempre que um sensor produz um pulso o CLP incrementa um contador que será chamado de contador1 e sempre que o outro sensor emite um pulso o CLP incrementa outro contador, que será chamado de contador2. Supondo que no sentido horário o parafuso passa primeiro pelo sensor do contador1, foi feita uma tabela para identificar o padrão que emerge na contagem, estando a bobinadeira começando a girar no sentido horário e mudando o sentido na segunda volta, e começando no sentido anti-horário e mudando de posição na segunda volta.

Começando no sentido horário

Tabela 3: comportamento do contador quando motor inicia no sentido horário e muda para o sentido anti-horário na segunda volta.

Contador1	0	1	1	2	2	2	3	3	4
Contador2	0	0	1	1	2	3	3	4	4

Começando no sentido anti-horário

Tabela 4: comportamento do contador quando motor inicia no sentido anti-horário e muda para o sentido horário na segunda volta.

Contador1	0	0	1	1	2	3	3	4	4
Contador2	0	1	1	2	2	2	3	3	4

Assim verifica-se que uma forma eficiente de saber se o eixo está no sentido horário ou anti-horário é sabendo qual contador está com o maior valor. Caso seja o contador1 o sistema está no sentido horário, caso seja o contador2 está no anti-horário. Sobretudo se o sistema não parar antes de uma volta completa este método será bastante eficaz. Supondo, também, uma variável de estados de dois bits, para definir três estados possíveis: inicial, horário e anti-horário. Seja esta uma variável de estados A, e seus *bits* A1 e A0, foi feita a tabela 5.

Tabela 5: tabela da verdade da variável de estados A

Estado	Bits da variável A	
	A1	A0
Inicial	0	0
Horário	0	1
Anti-horário	1	0
Proibido	1	1

A lógica poderia ser implementada da seguinte maneira, em C:

```
Evento(pulso em X0 ou X1)
if(contador1<contador2&&horario)
    A=anti-horario;
if(contador1>contador2&&anti-horario)
    A=horário;
    if(contador1>contador2&&inicial)
        A=horário;
    if(contador1<contador2&&inicial)
```

```

A=anti-horario;
if(A=horario)
    contador3++;
if(A=anti-horario)
    contador3--;

```

Tal código poderia ser implementado em LADDER, também, pois existem blocos comparadores e contadores na maioria dos compiladores de LADDER. Um exemplo de implementação está na Figura 8.

Onde A1 e A0 são variáveis internas de estado inicializadas com valor zero. O número de voltas é mostrado em um visor anexo que faz comunicação serial com o CLP.

- A entrada X2: está conectada ao botão verde da bobinaderia. Faz o motor parar. Isto é possível porque as entradas STF e STR do inversor estão conectadas ao CLP (saídas Y1 e Y2). Basta que se imponha o valor 0 para as saídas Y2 e Y1 para parar o motor (ou impor o valor 1 para ambas).
- As entradas X3 e X4: vêm de um pedal e são usadas para controlar o sentido da rotação (não confundir com o pedal conectado diretamente ao inversor, que controla a velocidade). Novamente isso é possível pois as entradas STF e STR do inversor estão conectadas ao CLP (saídas Y2 e Y1). Basta impor 1 para Y1 e 0 para Y2 para girar o motor em um sentido e trocar os bits para girar no sentido contrário
- As saídas Y2 e Y1: como já foi dito estão conectadas aos pinos STF e STR do inversor, e controlam se o motor vai estar no sentido horário, anti-horário ou parado. As saídas são função de X2, X3 e X4.
- Saída Y3: desativa o freio do motor.
- A saída Y0: está ligada a entrada RL/OH do inversor que normalmente é usada para desligar o inversor em caso de emergências.

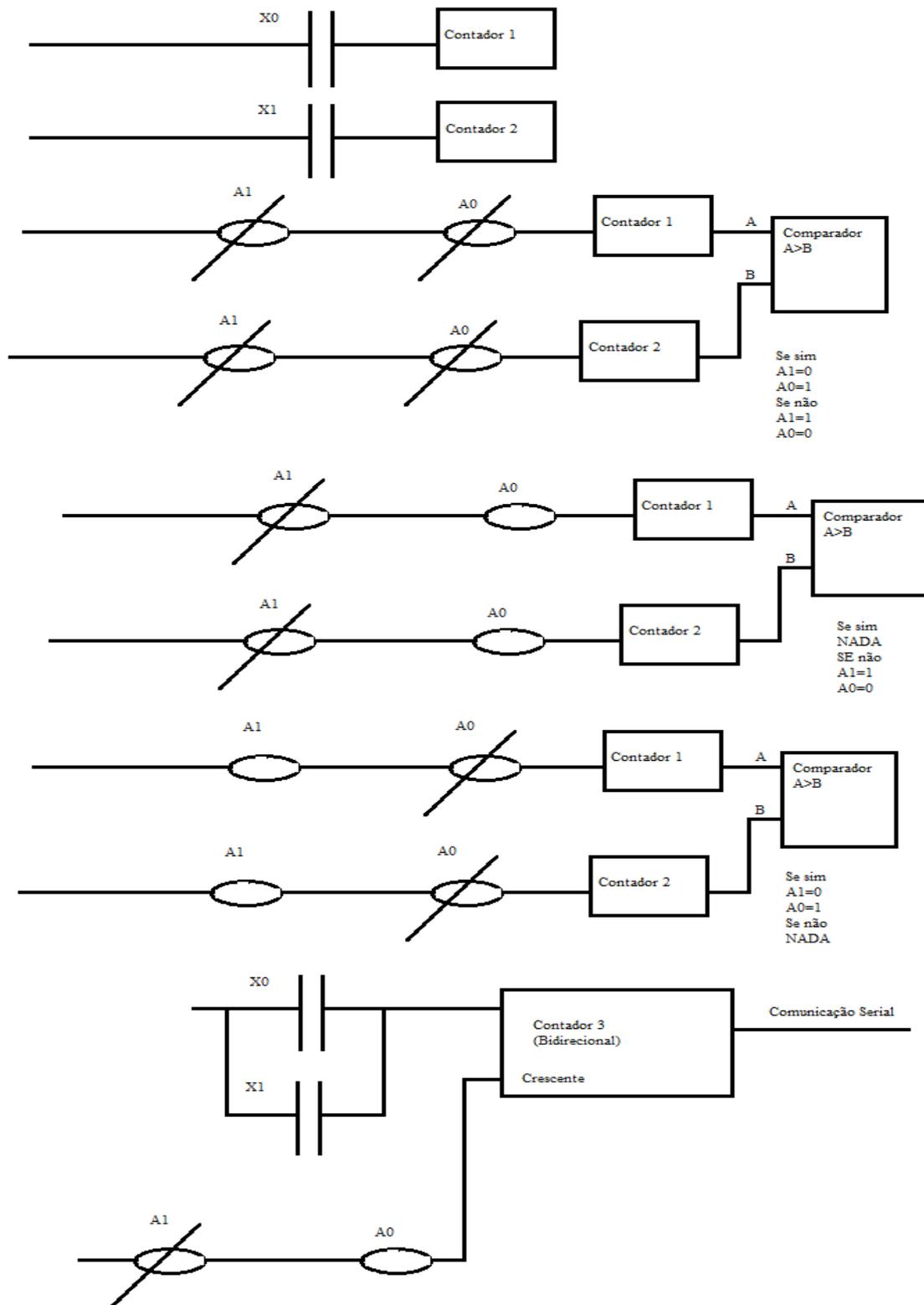


Figura 8: Exemplo de implementação em LADDER para a possível lógica da contagem de voltas.

O motor realmente precisa de uma tensão de 220V para liberar o freio e está conectado à saída Y3 do CLP. Entretanto não foi possível testar as demais hipóteses em relação ao CLP pois chegou-se à conclusão de que foi ele a peça que quebrou durante a conexão errada da bobinadeira. Quando ligou-se o CLP e mediu-se a tensão entre os

terminais 24V e 0V não foi obtido sinal algum de tensão. Conclui-se, então, que a fonte interna do CLP foi danificada.

Os demais equipamentos presentes na bobinadeira foram: disjuntor, do contactor e do *keypad*. O *keypad*, cuja especificação é Mitsubishi FX-10-DU, é um teclado acoplado a um visor, mostrado na Figura 9.



Figura 9: Keypad utilizado.

O *keypad* se conecta apenas ao CLP, de onde provém, inclusive sua alimentação. Comunica-se por protocolo RS232 e informa o número de voltas que se têm, solicita o número de voltas desejadas e envia para o CLP o comando de parada quando o valor desejado é atingido.

O disjuntor fica entre o contactor e o inversor e tem a finalidade de proteger o sistema contra sobrecorrentes.

A bobina do contactor é conectada ao botão vermelho, que, portanto, tem a finalidade de ligar e desligar a bobinadeira.

O motor é de freio, como já foi dito, e para desativar o freio é necessário impor uma tensão de 220V nos terminais do freio. Possivelmente foi isto que acarretou o dano ao sistema, pois a tensão do freio foi conseguida entre uma fase e o neutro. Como a bobinadeira foi conectada sem neutro o freio não destravou e a corrente de partida do motor permaneceu muito tempo a mais do que o CLP suportaria. O disjuntor, então, estava sobre dimensionado.

Quando o motor foi aberto, verificou-se que a ponte de diodos do freio estava queimada, devido ao surto de corrente. Trocada, então, a ponte de diodos e testou-se o freio, o qual verificou-se que funcionava.

2.6.2 O INVERSOR WEG

O inversor possui uma interface programável que nos permite ajustar o valor dos parâmetros internos do inversor, tais como: frequência, tempo de aceleração, tempo de

desaceleração, dentre outros. Para utilizar a interface homem-máquina é necessário acessar o parâmetro 000 e modificar seu valor para “5”. Para fazer isso precisa-se energizar o inversor. Então aparecerá no display de led’s a palavra “rdy” indicando que o equipamento foi energizado sem erros e está pronto para uso (“rdy” significa “ready”, isto é, “pronto” em inglês). Após isso verifica-se os leds com as identificações “value” e “parameter”, escritas ao lado. Se o led “parameter” estiver aceso, significa que o número mostrado no display é o parâmetro correspondente. Caso o led aceso seja o led identificado com “value”, o número mostrado é o valor do parâmetro. Para ser feita a permutação entre “value” e “parameter” deve-se apertar o botão “P”. O parâmetro ou valor podem ser alterados utilizando-se as teclas com desenhos de setas, para aumentar ou diminuir. Após a energização do inversor, escolhe-se o parâmetro 000 e estabelece-se seu valor em “5”.

Após o ajuste parâmetro 000, modificou-se o valor do parâmetro 229 para 0. Este é o ajuste que permite o controle da frequência de saída do inversor pelo teclado e não pelos bornes de controle do inversor, após o motor ser acionado. Também foi ajustada a frequência máxima de saída para 120 hertz, através do parâmetro 134.

Após ter-se todos os parâmetros de interesse ajustados convenientemente, pressiona-se a tecla “I/O” para o motor ser acionado. Uma vez acionado a frequência de saída do inversor se elevou até o patamar de 120 Hz. E depois foi diminuída para 60Hz, utilizando o teclado. Uma imagem do inversor utilizado está na Figura 10.



Figura 10: imagem do inversor WEG utilizado.

3 CONCLUSÃO

O estágio foi uma experiência de grande valor para o aprendiz, sobretudo por possibilitar uma visão de engenharia fora do âmbito acadêmico.

As atividades desempenhadas permitiram a consolidação dos conhecimentos de eletrônica, circuitos elétricos, máquinas elétricas, eletrônica de potência, controle e programação. Também foi crucial a tomada de consciência em relação ao que precisa ser aperfeiçoado e do que é praticado usualmente no mercado.

Foi enriquecedor conviver com técnicos e eletricitistas, como os que fizeram o curso de manutenção e instalação de painéis fotovoltaicos no SENAI, comerciantes de material e operários da fábrica.

Um ponto importante a destacar é que o engenheiro precisa estar atento às mudanças do mercado e aos novos equipamentos disponíveis, que precisam ser testados e, caso sejam mais eficientes, utilizados pela empresa. O engenheiro deve sempre manter-se atualizado, sempre há coisas novas a serem estudadas e processos a serem aperfeiçoados.

BIBLIOGRAFIA

MITSUBISHI, **Hardware Manual- FX Series Programmable controler**. Mitsubishi. Outubro. 2000

MITSUBISHI, **FX-10-DU-Operation user manual**. Mitsubishi. 1993

MITSUBISHI, **General-purpose inverter instruction manual FR-A44-0.4k to 3.7k** .Mitsubishi. Sem data.

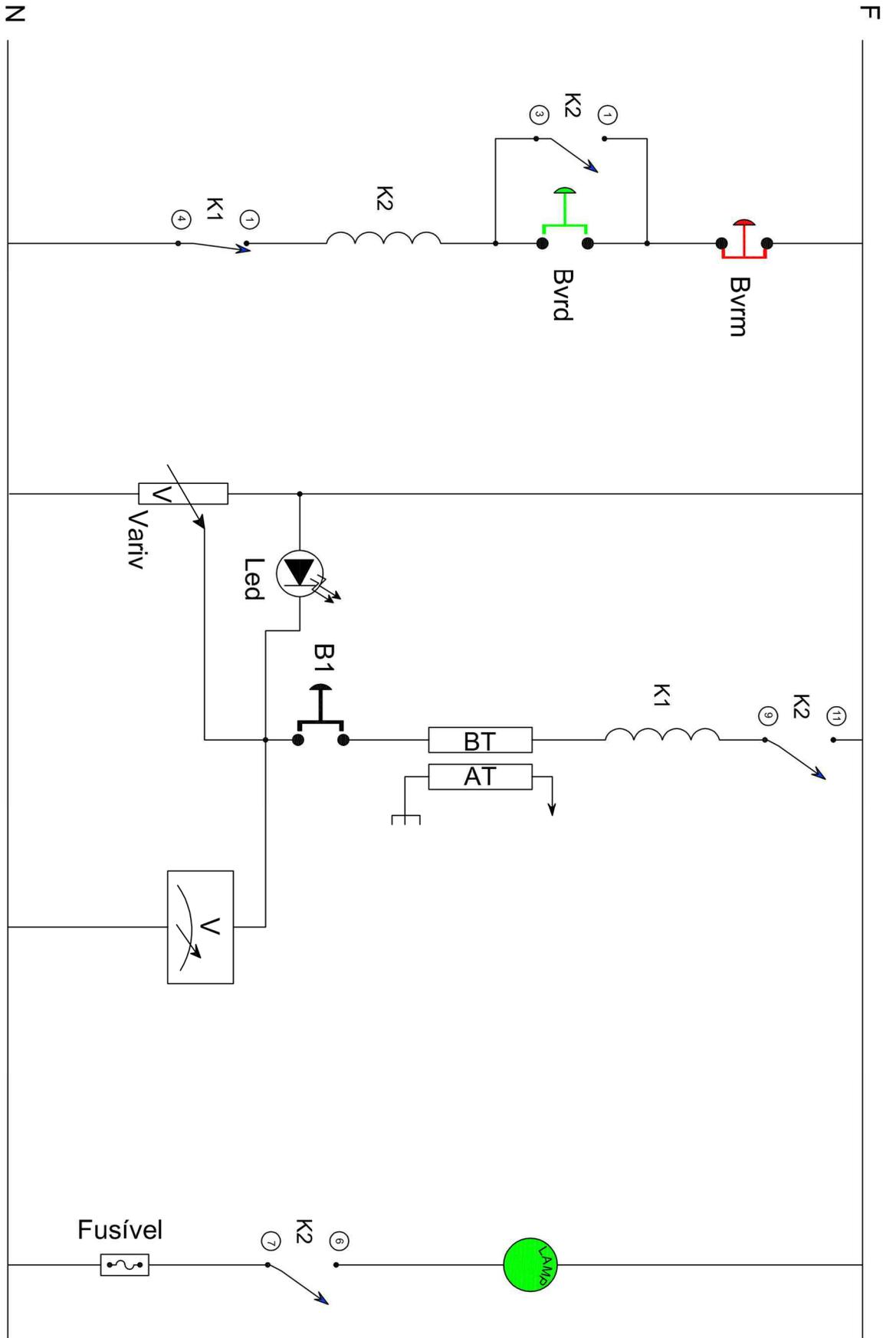
WEG, **Inversor de frecuencia CFW10 Manual do Usuário**. WEG. Maio .2013

FITZGERALD, A.E.; KINGSLEY JR, Charles; UMANS, Stephen D. **Máquinas elétricas**. McGraw Hill. Sétima Edição. 2014

HART, Daniel W. **Eletrônica de potência: análise e projetos de circuitos**. McGraw Hill. 2012.

SEDRA, Adel S.; SMITH, Kenneth C. **Microeletrônica**. Pearson Prentice Hall. Quinta edição. 2007.

OGATA, Katsuhiko. **Engenharia de controle moderno**. Pearson Prentice Hall. Quarta Edição. 2004.



APÊNDICE B- CÓDIGO EM C, PARA COMUNICAÇÃO ENTRE O PIC E A TELA DE LCD.

O código que permite a comunicação entre o PIC e o display de LCD é o seguinte. Foi adaptado do trabalho do técnico Lúcio Hélio Santana. É necessário que este código esteja salvo na pasta do código principal para que se possa escrever na tela de LCD usando as funções de escrita da linguagem C.

```

0014: MOVLW 41
0015: MOVWF 04
0016: BCF 03.7
0017: MOVF 00,W
0018: BTFSC 03.2
0019: GOTO 028
001A: MOVLW 0C
001B: MOVWF 78
001C: CLRF 77
001D: DECFSZ 77,F
001E: GOTO 01D
001F: DECFSZ 78,F
0020: GOTO 01C
0021: MOVLW F7
0022: MOVWF 77
0023: DECFSZ 77,F
0024: GOTO 023
0025: GOTO 026
0026: DECFSZ 00,F
0027: GOTO 01A
0028: RETURN
*
0050: MOVLW 01
0051: SUBWF 46,F
0052: BTFSS 03.0
0053: GOTO 060
0054: MOVLW 46
0055: MOVWF 04
0056: BCF 03.7
0057: MOVF 00,W
0058: BTFSC 03.2
0059: GOTO 060
005A: MOVLW 02
005B: MOVWF 77
005C: DECFSZ 77,F
005D: GOTO 05C
005E: DECFSZ 00,F
005F: GOTO 05A
0060: RETURN
.....
.....
.....

```

```

.....
/*=====
.....
Definição da substituição do "Identificado" pelo
"Texto" especificado.
.....
=====*/

.....
..... #define CS_LCD PIN_B2 // CS do LCD
----- (pino 6).
..... #define RS_LCD PIN_B0 // RS do LCD
----- (pino 4).
.....
..... #define D4_LCD PIN_D4 // D4 do
LCD ---- (pino 11).
..... #define D5_LCD PIN_D5 // D5 do
LCD ---- (pino 12).
..... #define D6_LCD PIN_D6 // D6 do
LCD ---- (pino 13).
..... #define D7_LCD PIN_D7 // D7 do
LCD ---- (pino 14).
.....
..... #define back_light PIN_D2 // Back
Light
.....
..... // RELE
..... #define RELE_1 PIN_B4 // RELE
.....
.....
/*=====
.....
INCLUSÃO DE BIBLIOTECAS - ARQUIVOS DE
DRIVERES
.....
=====*/

.....
..... #include <LCD_16x2_Lib.c> // Biblioteca de
manipulação do LCD 16x2.
.....
.....
/*=====
.....
=====
.....
..... ARQUIVO - DRIVER: LCD_16x2_LIB.C
..... DATA DE CRIAÇÃO: 29/05/2005
..... REVISÃO: 1.0
..... DATA DA ULTIMA REV: 29/05/2005
..... ULTIMA REV: 1.1
..... MICROCONTROLADOR: FAMILIA PIC
.....
..... FUNÇÃO:
.....
.....
..... DESCRIÇÃO: BIBLIOTECA DE MANIPULAÇÃO DO
DISPLAY DE CRISTAL LÍQUIDO (LCD 16X2).
.....
..... AUTOR: TÉC. LÚCIO HÉLIO
.....
.....
=====
.....
=====*/
.....
.....

```

```

.....
..... #ifndef CS_LCD
.....
..... #define CS_LCD PIN_E0 // Pino de Controle "CS -
CHIP SELECT" do LCD ----- (pino 6).
..... #define RS_LCD PIN_E1 // Pino de Controle "RS -
REGISTER SELECT" do LCD ----- (pino 4).
..... #define D4_LCD PIN_D4 // Pino do Barramento de
Dados "D4 - Dado 4" do LCD --- (pino 11).
..... #define D4_LCD PIN_D5 // Pino do Barramento de
Dados "D5 - Dado 5" do LCD --- (pino 12).
..... #define D4_LCD PIN_D6 // Pino do Barramento de
Dados "D6 - Dado 6" do LCD --- (pino 13).
..... #define D4_LCD PIN_D7 // Pino do Barramento de
Dados "D7 - Dado 7" do LCD --- (pino 14).
.....
..... #endif
.....
..... #define lcd_type 2 // 0=5x7, 1=5x10, 2=2
linhas
..... #define lcd_seg_lin 0x40 // Endereço da segunda
linha na RAM do LCD
.....
.....
..... // A constante abaixo define a seqüência de
inicialização do LCD.
..... byte CONST seq_ini_lcd[4] = {0x20 | (lcd_type << 2), 0x0c,
1, 6};
.....
..... byte lcd_linha;
.....
.....
/*=====
.....
..... FUNÇÃO: ENVIA_NIBBLE_LCD()
..... PARÂMETROS: BYTE, DADO...
..... RETORNO: BYTE, NENHUM.
.....
..... DESCRIÇÃO: ENVIA UM DADO DE QUATRO BIT´S PARA O
BARRAMENTO DO DISPLAY.
.....
.....
=====*/
..... void envia_nibble_lcd( byte dado )
..... {
.....
..... output_bit(D4_LCD,bit_test(dado,0)); // Coloca os
quatro bits nas saídas.
*
0029: BTFSC 47.0
002A: GOTO 02D
002B: BCF 08.4
002C: GOTO 02E
002D: BSF 08.4
002E: BSF 03.5
002F: BCF 08.4
..... output_bit(D5_LCD,bit_test(dado,1));
0030: BCF 03.5
0031: BTFSC 47.1
0032: GOTO 035
0033: BCF 08.5

```

```

0034: GOTO 036
0035: BSF 08.5
0036: BSF 03.5
0037: BCF 08.5
.....
output_bit(D6_LCD,bit_test(dado,2));
0038: BCF 03.5
0039: BTFSC 47.2
003A: GOTO 03D
003B: BCF 08.6
003C: GOTO 03E
003D: BSF 08.6
003E: BSF 03.5
003F: BCF 08.6
.....
output_bit(D7_LCD,bit_test(dado,3));
0040: BCF 03.5
0041: BTFSC 47.3
0042: GOTO 045
0043: BCF 08.7
0044: GOTO 046
0045: BSF 08.7
0046: BSF 03.5
0047: BCF 08.7
.....
..... output_high(CS_LCD); // Dá um pulso no pino de
Enable (CS).
0048: BCF 06.2
0049: BCF 03.5
004A: BSF 06.2
.....
output_low(CS_LCD);
004B: BSF 03.5
004C: BCF 06.2
004D: BCF 03.5
004E: BCF 06.2
..... }
004F: RETURN
.....
.....
.....
/*=====
.....
..... FUNÇÃO: ENVIA_BYTE_LCD()
..... PARÂMETROS: BYTE, ENDEREÇO, DADO...
..... RETORNO: BYTE, NENHUM.
.....
..... DESCRIÇÃO: ENVIA UM DADO DE QUATRO BIT´S PARA O
BARRAMENTO DO DISPLAY.
.....
.....
=====*/
..... void envia_byte_lcd( boolean endereco, byte dado )
..... {
.....
..... output_low(RS_LCD); // Coloca o pino "RS" em "0".
*
0061: BSF 03.5
0062: BCF 06.0
0063: BCF 03.5
0064: BCF 06.0
.....
..... // Aguarda o display ficar
desocupado.

```

```

..... //while ( bit_test(lcd_le_byte(),7) ) ;
.....
..... // Configura o pino "RS"
dependendo do modo selecionado.
..... output_bit(RS_LCD,endereco);
0065: MOVF 44,F
0066: BTFSS 03.2
0067: GOTO 06A
0068: BCF 06.0
0069: GOTO 06B
006A: BSF 06.0
006B: BSF 03.5
006C: BCF 06.0
.....
..... delay_us(100); // Aguarda 100 us.
006D: MOVLW 64
006E: BCF 03.5
006F: MOVWF 46
0070: CALL 050
.....
..... // caso a linha rw esteja definida, coloca em "0".
..... // output_low(RW_LCD);
.....
..... // Desabilita o Display.
..... output_low(CS_LCD);
0071: BSF 03.5
0072: BCF 06.2
0073: BCF 03.5
0074: BCF 06.2
.....
..... envia_nibble_lcd(dado >> 4); // Envia a primeira
parte do byte.
0075: SWAPF 45,W
0076: MOVWF 46
0077: MOVLW 0F
0078: ANDWF 46,F
0079: MOVF 46,W
007A: MOVWF 47
007B: CALL 029
.....
..... envia_nibble_lcd(dado & 0x0f); // Envia a segunda
parte do byte.
007C: MOVF 45,W
007D: ANDLW 0F
007E: MOVWF 46
007F: MOVWF 47
0080: CALL 029
.....
..... }
0081: RETURN
.....
.....
.....
.....
/*=====
.....
..... FUNÇÃO: INI_LCD()
..... PARÂMETROS: BYTE, NENHUM.
..... RETORNO: BYTE, NENHUM.
.....
..... DESCRIÇÃO: ROTINA DE INICIALIZAÇÃO DO LCD.
.....
.....

```

```

.....
=====*/
..... void ini_lcd_16x2()
..... {
.....     byte conta;
.....
.....     output_low(CS_LCD);
0082: BSF    03.5
0083: BCF    06.2
0084: BCF    03.5
0085: BCF    06.2
.....     output_low(D4_LCD);
0086: BSF    03.5
0087: BCF    08.4
0088: BCF    03.5
0089: BCF    08.4
.....     output_low(D6_LCD);
008A: BSF    03.5
008B: BCF    08.6
008C: BCF    03.5
008D: BCF    08.6
.....     output_low(D7_LCD);
008E: BSF    03.5
008F: BCF    08.7
0090: BCF    03.5
0091: BCF    08.7
.....     output_low(RS_LCD);
0092: BSF    03.5
0093: BCF    06.0
0094: BCF    03.5
0095: BCF    06.0
.....
.....     // output_high(RW_LCD);
.....
.....     output_low(CS_LCD);
0096: BSF    03.5
0097: BCF    06.2
0098: BCF    03.5
0099: BCF    06.2
.....
.....     delay_ms(15);
009A: MOVLW  0F
009B: MOVWF   41
009C: CALL    014
.....
.....     // Eenvia uma seqüência de 3 vezes ovalor "0x03" e
depois o valor "0x02" para // configurar o barramento do LCD no modo de 4 bits.
.....     for(conta = 1 ;conta <= 3 ; ++conta)
009D: MOVLW  01
009E: MOVWF   33
009F: MOVF   33,W
00A0: SUBLW  03
00A1: BTFSS  03.0
00A2: GOTO   0AB
.....     {
.....         envia_nibble_lcd(3);
00A3: MOVLW  03
00A4: MOVWF   47
00A5: CALL    029
.....         delay_ms(5);
00A6: MOVLW  05

```

```

00A7: MOVWF 41
00A8: CALL 014
..... }
00A9: INCF 33,F
00AA: GOTO 09F
.....
..... envia_nibble_lcd(2);
00AB: MOVLW 02
00AC: MOVWF 47
00AD: CALL 029
.....
..... // Envia uma "string" de inicialização do Display.
..... for(conta = 0; conta <=3 ; ++conta)
envia_byte_lcd(0,seq_ini_lcd[conta]);
00AE: CLRF 33
00AF: MOVF 33,W
00B0: SUBLW 03
00B1: BTFSS 03.0
00B2: GOTO 0BC
00B3: MOVF 33,W
00B4: CALL 004
00B5: MOVWF 34
00B6: CLRF 44
00B7: MOVF 34,W
00B8: MOVWF 45
00B9: CALL 061
00BA: INCF 33,F
00BB: GOTO 0AF
..... }
00BC: BCF 0A.3
00BD: BCF 0A.4
00BE: GOTO 5CD (RETURN)
.....
.....
.....
.....
/*=====
.....
..... FUNÇÃO: XY_LCD()
..... PARÂMETROS: BYTE, X,Y.
..... RETORNO: BYTE, NENHUM.
.....
..... DESCRIÇÃO: ROTINA DE...
.....
=====*/
..... void posicao_xy_lcd( byte x, byte y)
..... {
.....
..... BYTE address;
.....
..... switch(y) {
00BF: MOVLW 01
00C0: SUBWF 42,W
00C1: ADDLW FC
00C2: BTFSC 03.0
00C3: GOTO 0D1
00C4: ADDLW 04
00C5: GOTO 0D9
..... case 1 : address=0x80;break;
00C6: MOVLW 80
00C7: MOVWF 43

```

```

00C8: GOTO 0D1
..... case 2 : address=0xc0;break;
00C9: MOVLW C0
00CA: MOVWF 43
00CB: GOTO 0D1
..... case 3 : address=0x94;break;
00CC: MOVLW 94
00CD: MOVWF 43
00CE: GOTO 0D1
..... case 4 : address=0xd4;break;
00CF: MOVLW D4
00D0: MOVWF 43
..... }
..... address+=x-1;
00D1: MOVLW 01
00D2: SUBWF 41,W
00D3: ADDWF 43,F
..... envia_byte_lcd(0,address);
00D4: CLRF 44
00D5: MOVF 43,W
00D6: MOVWF 45
00D7: CALL 061
..... }
00D8: RETURN
.....
.....
.....
.....
/*=====
.....
..... FUNÇÃO: EXIBE_LCD()
..... PARÂMETROS: BYTE, C...
..... RETORNO: BYTE, NENHUM.
.....
..... DESCRIÇÃO: ROTINA PARAEXIBIR TEXTOS (CARACTERES) E
COMANDOS.
.....
.....
=====*/
..... void exhibe_lcd( char character)
..... {
..... switch (character)
..... {
*
00E1: MOVF 40,W
00E2: XORLW 0C
00E3: BTFSC 03.2
00E4: GOTO 0EC
00E5: XORLW 06
00E6: BTFSC 03.2
00E7: GOTO 0F4
00E8: XORLW 02
00E9: BTFSC 03.2
00EA: GOTO 0FA
00EB: GOTO 0FF
..... case '\f' : // Envia um byte de Comando de
Limpar (CLEAR) o Display.
.....
..... envia_byte_lcd(0,1);
00EC: CLRF 44
00ED: MOVLW 01
00EE: MOVWF 45

```



```

..... xy_lcd(x,y); // Seleciona a posição do
caractere.
.....
..... output_high(RS_LCD); // Ativa o pino "RS".
.....
..... valor = lcd_le_byte(); // Lê o caractere.
.....
..... output_low(RS_LCD); // Desativa o pino "RS".
.....
..... return valor; // Retorna o valor do caractere.
.....
..... }
..... */
.....
.....
//=====
.....
.....
.....
.....
/*=====
.....
..... FUNÇÃO: VOID MAIN()
..... PARÂMETROS: NENHUM
..... RETORNO: NENHUM
.....
..... DESCRIÇÃO: PARTE PRINCIPAL DO PROGRAMA...
.....
.....
=====*/
.....
..... void main() // Inicio do
programa principal.
..... {
*
05B9: CLRF 04
05BA: BCF 03,7
05BB: MOVLW 1F
05BC: ANDWF 03,F
05BD: BSF 03,5
05BE: BSF 1F,0
05BF: BSF 1F,1
05C0: BSF 1F,2
05C1: BCF 1F,3
05C2: MOVLW 07
05C3: MOVWF 1C
..... // Cria uma variável local
"valor_ad" do tipo long int.
..... long int valor_ad;
.....
..... // int32 temp, aux, R, tensao;
..... float temp, aux, R, tensao;
..... // Cria uma variável local "temp" do tipo int32.
.....
..... delay_ms(1000);
05C4: MOVLW 04
05C5: BCF 03,5
05C6: MOVWF 33
05C7: MOVLW FA
05C8: MOVWF 41
05C9: CALL 014
05CA: DECFSZ 33,F

```

```

05CB: GOTO 5C7
.....
..... ini_lcd_16x2();
// Inicializa o LCD.
05CC: GOTO 082
.....
..... output_high(back_light);
// Liga o Back light.
05CD: BSF 03.5
05CE: BCF 08.2
05CF: BCF 03.5
05D0: BSF 08.2
.....
..... printf (exibe_lcd,"\f Temp");
05D1: MOVLW 0C
05D2: BSF 03.6
05D3: MOVWF 0D
05D4: MOVLW 00
05D5: MOVWF 0F
05D6: BCF 03.6
05D7: GOTO 105
.....
..... // - - - - -
- - - - -
.....
..... while(true) // Inicio da
estrutura de repetição.
..... {
.....
..... //-----< CONFIGURAÇÃO DO ADC >-----
.....
.....
..... setup_ADC_ports (RA0_analog); // CONFIGURA O PINO
"RA0" ENTRADA ANALÓGICA.
05D8: BSF 03.5
05D9: BCF 1F.0
05DA: BSF 1F.1
05DB: BSF 1F.2
05DC: BSF 1F.3
.....
..... setup_adc(ADC_CLOCK_DIV_8); // VELOCIDADE = Fosc/8.
05DD: BCF 1F.6
05DE: BCF 03.5
05DF: BSF 1F.6
05E0: BCF 1F.7
05E1: BSF 03.5
05E2: BSF 1F.7
05E3: BCF 03.5
05E4: BSF 1F.0
.....
..... set_adc_channel(0); // CANAL "0".
05E5: MOVLW 00
05E6: MOVWF 78
05E7: MOVF 1F,W
05E8: ANDLW C7
05E9: IORWF 78,W
05EA: MOVWF 1F
.....
..... delay_us(10); // DELAY PARA
ESTABILIZAÇÃO.
05EB: MOVLW 20
05EC: MOVWF 77
05ED: DECFSZ 77,F

```

```

05EE: GOTO 5ED
05EF: GOTO 5F0
05F0: NOP
.....
.....
..... //-----
-----
.....
.....          valor_ad = read_adc();          // Lê o
resultado da conversão A/D.
05F1: BSF 1F.2
05F2: BTFSC 1F.2
05F3: GOTO 5F2
05F4: BSF 03.5
05F5: MOVF 1E,W
05F6: BCF 03.5
05F7: MOVWF 21
05F8: MOVF 1E,W
05F9: MOVWF 22
.....
.....
..... // - - - - -
- - - - -
..... delay_us(100);
05FA: MOVLW 64
05FB: MOVWF 46
05FC: CALL 050
..... //
.....          //temp = (1000 * (int32)valor_ad) / 1023;
.....          // temp = (int32)valor_ad*100 ;
.....          tensao=((float) valor_ad*0.0048875855327468);
05FD: MOVF 22,W
05FE: MOVWF 34
05FF: MOVF 21,W
0600: MOVWF 33
0601: GOTO 152
0602: MOVF 77,W
0603: MOVWF 33
0604: MOVF 78,W
0605: MOVWF 34
0606: MOVF 79,W
0607: MOVWF 35
0608: MOVF 7A,W
0609: MOVWF 36
060A: MOVWF 43
060B: MOVF 79,W
060C: MOVWF 42
060D: MOVF 78,W
060E: MOVWF 41
060F: MOVF 77,W
0610: MOVWF 40
0611: MOVLW 0A
0612: MOVWF 47
0613: MOVLW 28
0614: MOVWF 46
0615: MOVLW 20
0616: MOVWF 45
0617: MOVLW 77
0618: MOVWF 44
0619: CALL 171
061A: MOVF 7A,W
061B: MOVWF 32

```

```

061C: MOVF 79,W
061D: MOVWF 31
061E: MOVF 78,W
061F: MOVWF 30
0620: MOVF 77,W
0621: MOVWF 2F
..... // aux=36000-40000*tensao;
..... // temp=aux;
..... R=((900*tensao)/(12-tensao));
0622: CLRF 43
0623: CLRF 42
0624: MOVLW 61
0625: MOVWF 41
0626: MOVLW 88
0627: MOVWF 40
0628: MOVF 32,W
0629: MOVWF 47
062A: MOVF 31,W
062B: MOVWF 46
062C: MOVF 30,W
062D: MOVWF 45
062E: MOVF 2F,W
062F: MOVWF 44
0630: CALL 171
0631: MOVF 77,W
0632: MOVWF 33
0633: MOVF 78,W
0634: MOVWF 34
0635: MOVF 79,W
0636: MOVWF 35
0637: MOVF 7A,W
0638: MOVWF 36
0639: BSF 03.1
063A: CLRF 3A
063B: CLRF 39
063C: MOVLW 40
063D: MOVWF 38
063E: MOVLW 82
063F: MOVWF 37
0640: MOVF 32,W
0641: MOVWF 3E
0642: MOVF 31,W
0643: MOVWF 3D
0644: MOVF 30,W
0645: MOVWF 3C
0646: MOVF 2F,W
0647: MOVWF 3B
0648: CALL 1E6
0649: MOVF 36,W
064A: MOVWF 3A
064B: MOVF 35,W
064C: MOVWF 39
064D: MOVF 34,W
064E: MOVWF 38
064F: MOVF 33,W
0650: MOVWF 37
0651: MOVF 7A,W
0652: MOVWF 3E
0653: MOVF 79,W
0654: MOVWF 3D
0655: MOVF 78,W
0656: MOVWF 3C

```

```

0657: MOVF 77,W
0658: MOVWF 3B
0659: GOTO 327
065A: MOVF 7A,W
065B: MOVWF 2E
065C: MOVF 79,W
065D: MOVWF 2D
065E: MOVF 78,W
065F: MOVWF 2C
0660: MOVF 77,W
0661: MOVWF 2B
..... temp=2.5974025974025974025974026*R-
259.74025974025974025974026;
0662: MOVLW D8
0663: MOVWF 43
0664: MOVLW 3B
0665: MOVWF 42
0666: MOVLW 26
0667: MOVWF 41
0668: MOVLW 80
0669: MOVWF 40
066A: MOVF 2E,W
066B: MOVWF 47
066C: MOVF 2D,W
066D: MOVWF 46
066E: MOVF 2C,W
066F: MOVWF 45
0670: MOVF 2B,W
0671: MOVWF 44
0672: CALL 171
0673: MOVF 77,W
0674: MOVWF 33
0675: MOVF 78,W
0676: MOVWF 34
0677: MOVF 79,W
0678: MOVWF 35
0679: MOVF 7A,W
067A: MOVWF 36
067B: BSF 03.1
067C: MOVF 7A,W
067D: MOVWF 3A
067E: MOVF 79,W
067F: MOVWF 39
0680: MOVF 78,W
0681: MOVWF 38
0682: MOVF 77,W
0683: MOVWF 37
0684: MOVLW C1
0685: MOVWF 3E
0686: MOVLW DE
0687: MOVWF 3D
0688: MOVLW 01
0689: MOVWF 3C
068A: MOVLW 87
068B: MOVWF 3B
068C: CALL 1E6
068D: MOVF 7A,W
068E: MOVWF 26
068F: MOVF 79,W
0690: MOVWF 25
0691: MOVF 78,W
0692: MOVWF 24

```

```

0693: MOVF 77,W
0694: MOVWF 23
.....
..... if(temp<=105)
0695: MOVF 26,W
0696: MOVWF 36
0697: MOVF 25,W
0698: MOVWF 35
0699: MOVF 24,W
069A: MOVWF 34
069B: MOVF 23,W
069C: MOVWF 33
069D: CLRF 3A
069E: CLRF 39
069F: MOVLW 52
06A0: MOVWF 38
06A1: MOVLW 85
06A2: MOVWF 37
06A3: CALL 3F3
06A4: BTFSC 03.0
06A5: GOTO 6A8
06A6: BTFSS 03.2
06A7: GOTO 6AD
..... output_high(RELE_1);
06A8: BSF 03.5
06A9: BCF 06.4
06AA: BCF 03.5
06AB: BSF 06.4
..... else if(temp>105)
06AC: GOTO 6C2
06AD: CLRF 36
06AE: CLRF 35
06AF: MOVLW 52
06B0: MOVWF 34
06B1: MOVLW 85
06B2: MOVWF 33
06B3: MOVF 26,W
06B4: MOVWF 3A
06B5: MOVF 25,W
06B6: MOVWF 39
06B7: MOVF 24,W
06B8: MOVWF 38
06B9: MOVF 23,W
06BA: MOVWF 37
06BB: CALL 3F3
06BC: BTFSS 03.0
06BD: GOTO 6C2
..... output_low(RELE_1);
06BE: BSF 03.5
06BF: BCF 06.4
06C0: BCF 03.5
06C1: BCF 06.4
.....
.....
..... seguinte forma:
..... // O escalonamento é realizado da
..... // resultado = (1000 * valor lido)
..... // Repare que é necessário
..... // equação para 32 bits para que o
..... / 1023
..... converter a segunda parte da
..... compilador efetue o

```

```

..... // cálculo corretamente.
.....
..... // -----
.....
.....
..... posicao_xy_lcd(8,1);
06C2: MOVLW 08
06C3: MOVWF 41
06C4: MOVLW 01
06C5: MOVWF 42
06C6: CALL 0BF
.....
..... printf (exibe_lcd,"%f C ",temp); // imprime o valor
da tensão no display
06C7: MOVLW 89
06C8: MOVWF 04
06C9: MOVF 26,W
06CA: MOVWF 36
06CB: MOVF 25,W
06CC: MOVWF 35
06CD: MOVF 24,W
06CE: MOVWF 34
06CF: MOVF 23,W
06D0: MOVWF 33
06D1: MOVLW 02
06D2: MOVWF 37
06D3: GOTO 481
06D4: MOVLW 11
06D5: BSF 03.6
06D6: MOVWF 0D
06D7: MOVLW 00
06D8: MOVWF 0F
06D9: BCF 03.0
06DA: MOVLW 05
06DB: BCF 03.6
06DC: MOVWF 33
06DD: GOTO 563
.....
..... } // Fim da
estrutura de repetição.
06DE: GOTO 5D8
..... } // Fim do
programa principal.
.....
.....
//=====
06DF: SLEEP

```

Configuration Fuses:

Word 1: 3F3A HS NOWDT NOPUT NOBROWNOUT NOLVP NOCPD NOWRT NODEBUG NOPROTECT

APÊNDICE C-CÓDIGO EM C, PARA O CONTROLE DA TEMPERATURA.

Código do controle de temperatura:

```
#define CS_LCD    PIN_B2                // CS do LCD ----- (pino 6).
#define RS_LCD    PIN_B0                // RS do LCD ----- (pino 4).

#define D4_LCD    PIN_D4                // D4 do LCD ---- (pino 11).
#define D5_LCD    PIN_D5                // D5 do LCD ---- (pino 12).
#define D6_LCD    PIN_D6                // D6 do LCD ---- (pino 13).
#define D7_LCD    PIN_D7                // D7 do LCD ---- (pino 14).

#define back_light    PIN_D2            // Back Light

// RELE
#define RELE_1    PIN_B4                // RELE

/*=====
      INCLUSÃO DE BIBLIOTECAS - ARQUIVOS DE DRIVERES
=====*/

#include <LCD_16x2_Lib.c>                // Biblioteca de manipulação do LCD 16x2.

/*=====

FUNÇÃO:      VOID MAIN()
PARÂMETROS:  NENHUM
RETORNO:     NENHUM

DESCRIÇÃO:   PARTE PRINCIPAL DO PROGRAMA...

=====*/

void main()                               // Início do programa principal.
{
    // Cria uma variável local "valor_ad" do tipo long int.
    long int valor_ad;

    // int32    temp, aux, R, tensao;
    float temp, aux, R, tensao;
    // Cria uma variável local "temp" do tipo int32.

    delay_ms(1000);

    ini_lcd_16x2();                        // Inicializa o LCD.
    output_high(back_light);              // Liga o Back light.
    printf (exibe_lcd, "\f Temp");

    // - - - - -
```

```

while(true) // Inicio da estrutura de repetição.
{
//-----< CONFIGURAÇÃO DO ADC >-----

setup_ADC_ports (RA0_analog); // CONFIGURA O PINO "RA0" ENTRADA ANALÓGICA.
setup_adc(ADC_CLOCK_DIV_8); // VELOCIDADE = Fosc/8.
set_adc_channel(0); // CANAL "0".

delay_us(10); // DELAY PARA ESTABILIZAÇÃO.

//-----

valor_ad = read_adc(); // Lê o resultado da conversão A/D.

// - - - - -
delay_us(100);
//
//temp = (1000 * (int32)valor_ad) / 1023;
// temp = (int32)valor_ad*100 ;
tensao=((float) valor_ad*0.0048875855327468);
// aux=36000-40000*tensao;
// temp=aux;
R=((900*tensao)/(12-tensao));
temp=2.5974025974025974025974025974026*R-259.74025974025974025974026;

if(temp<=105)
output_high(RELE_1);
else if(temp>105)
output_low(RELE_1);

// O escalonamento é realizado da seguinte forma:
// resultado = (1000 * valor lido) / 1023
// Repare que é necessário converter a segunda parte da
// equação para 32 bits para que o compilador efetue o
// cálculo corretamente.

// - - - - -

posicao_xy_lcd(8,1);

printf (exibe_lcd,"%f C ",temp); // imprime o valor da tensão no display

} // Fim da estrutura de repetição.
} // Fim do programa principal.

```

APÊNDICE D-CÓDIGO EM VBA DAS MACROS UTILIZADAS NA PLANILHA EXCEL.

Código em VBA dos macros utilizados na planilha de tarifa elétrica:

Macro do gráfico da demanda fora de ponta(transforma um gráfico de linhas aleatório com doze rótulos horizontais no gráfico da demanda fora de ponta)

```
Sub demanda_fponta()
    ActiveSheet.ChartObjects("Gráfico 2").Activate
    ActiveChart.PlotArea.Select
    Call temp
    ActiveSheet.ChartObjects("Gráfico 2").Activate
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection(1).Name = "=Plan1!$C$25"
    ActiveChart.SeriesCollection(1).Values = "=Plan1!$C$26:$C$37"
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection(2).Name = "=Plan1!$L$25"
    ActiveChart.SeriesCollection(2).Values = "=Plan1!$L$26:$L$37"

End Sub
```

Macro do gráfico da demanda reativa (transforma um gráfico de linhas aleatório com doze rótulos horizontais no gráfico da demanda reativa)

```
Sub demanda_reativa()
    ActiveSheet.ChartObjects("Gráfico 2").Activate
    ActiveChart.PlotArea.Select
    Call temp
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection(1).Name = "=Plan1!$Q$25"
```

```

ActiveChart.SeriesCollection(1).Values = "=Plan1!$Q$26:$Q$37"
End Sub

```

Macro do gráfico do consumo de ponta (transforma um gráfico de linhas aleatório com doze rótulos horizontais no gráfico do consumo de ponta)

```

Sub Consumo_ponta()
ActiveSheet.ChartObjects("Gráfico 2").Activate
ActiveChart.PlotArea.Select
Call temp
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection(1).Name = "=Plan1!$G$25"
ActiveChart.SeriesCollection(1).Values = "=Plan1!$G$26:$G$37"
End Sub

```

Sub-rotina utilizada para apagar as entradas de legendas

```

Public Sub temp()
ActiveSheet.ChartObjects("Gráfico 2").Activate
For Each s In ActiveChart.SeriesCollection
s.Delete
Next s
End Sub

```

Macro do gráfico do consumo fora de ponta (transforma um gráfico de linhas aleatório com doze rótulos horizontais no gráfico do consumo fora de ponta)

```

Sub consumo_fponta()
ActiveSheet.ChartObjects("Gráfico 2").Activate
ActiveChart.PlotArea.Select
Call temp
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection(1).Name = "=Plan1!$F$25"

```

```
ActiveChart.SeriesCollection(1).Values = "=Plan1!$F$26:$F$37"
```

```
End Sub
```

Macro do gráfico da demanda de ponta (transforma um gráfico de linhas aleatório com doze rótulos horizontais no gráfico da demanda de ponta)

```
Sub demanda_ponta()
```

```
ActiveSheet.ChartObjects("Gráfico 2").Activate
```

```
ActiveChart.PlotArea.Select
```

```
Call temp
```

```
ActiveChart.SeriesCollection.NewSeries
```

```
ActiveChart.SeriesCollection(1).Name = "=Plan1!$B$25"
```

```
ActiveChart.SeriesCollection(1).Values = "=Plan1!$B$26:$B$37"
```

```
ActiveChart.SeriesCollection.NewSeries
```

```
ActiveChart.SeriesCollection(2).Name = "=Plan1!$M$25"
```

```
ActiveChart.SeriesCollection(2).Values = "=Plan1!$M$26:$M$37"
```

```
End Sub
```

Macro do gráfico dos custos (transforma um gráfico de linhas aleatório com doze rótulos horizontais no gráfico dos custos)

```
Sub custos()
```

```
ActiveSheet.ChartObjects("Gráfico 2").Activate
```

```
ActiveChart.PlotArea.Select
```

```
Call temp
```

```
ActiveChart.SeriesCollection.NewSeries
```

```
ActiveChart.SeriesCollection(1).Name = "=Plan1!$E$42"
```

```
ActiveChart.SeriesCollection(1).Values = "=Plan1!$E$43:$E$54"
```

```
ActiveChart.SeriesCollection.NewSeries
```

```
ActiveChart.SeriesCollection(2).Name = "=Plan1!$N$42"
```

```
ActiveChart.SeriesCollection(2).Values = "=Plan1!$N$43:$N$54"
```

```
ActiveChart.SeriesCollection.NewSeries
```

```
ActiveChart.SeriesCollection(3).Name = "=Plan1!$T$42"
```

```
ActiveChart.SeriesCollection(3).Values = "=Plan1!$T$43:$T$54"
```

```
ActiveChart.SeriesCollection(2).Select
With Selection.Format.Line
    .Visible = msoTrue
    .ForeColor.ObjectThemeColor = msoThemeColorAccent1
    .ForeColor.TintAndShade = 0
    .ForeColor.Brightness = 0
    .Transparency = 0
End With
ActiveChart.SeriesCollection(1).Select
With Selection.Format.Line
    .Visible = msoTrue
    .ForeColor.ObjectThemeColor = msoThemeColorAccent3
    .ForeColor.TintAndShade = 0
    .ForeColor.Brightness = 0
    .Transparency = 0
End With
ActiveChart.PlotArea.Select
ActiveChart.PlotArea.Select
ActiveChart.SeriesCollection(3).Select
With Selection.Format.Line
    .Visible = msoTrue
    .ForeColor.ObjectThemeColor = msoThemeColorBackground1
    .ForeColor.TintAndShade = 0
    .ForeColor.Brightness = -0.5
    .Transparency = 0
End With
End Sub
```

APÊNDICE E-TABELAS UTILIZADAS NA PLANILHA

EXCEL.

Tabela Principal

Mês	Dem.reg.ponta(kW)	Dem.reg.fp(kW)	dem. Reg. maior(kW)	Ultrap.(kW)	Consumo f.ponta(kW.h)	Consumo ponta(kW.h)	Con intermed
1	100	410	410	-90	10000	800	3
2	200	495	495	-5	20000	1000	3
3	200	500	500	0	15000	800	3
4	200	540	540	40	13000	2000	3
5	220	540	540	40	20000	800	3
6	300	530	530	30	10000	3000	3
7	150	500	500	0	30000	1500	0
8	310	500	500	0	12000	800	3
9	220	500	500	0	13000	800	3
10	110	478	478	-22	14000	4000	5
11	250	480	480	-20	17000	800	4
12	300	500	500	0	10000	800	3
		demP contratada	demFP contratada				
		220	500				

Tabela da Tarifa Verde

TARIFA VERDE				
mês	P.Dem.(R\$)	P.Cons.(R\$)	P.Multa(R\$)	Total Verde(R\$)
1	2555	2675	0	5230
2	2555	4894	0	7449
3	2555	3708,5	0	6263,5
4	2759,4	4207,1	408,8	7375,3
5	2759,4	4742	408,8	7910,2
6	2708,3	4347	306,6	7361,9
7	2555	7341	0	9896
8	2555	3088,4	0	5643,4
9	2555	3295,1	0	5850,1
10	2555	5933,8	0	8488,8
11	2555	4121,9	0	6676,9
12	2555	2675	0	5230

Tabela da Tarifa Azul

TARIFA AZUL							
mês	P.Dem.ponta(R\$)	P.Dem.fpo nta(R\$)	P.Consumo ponta(R\$)	P.Consum o.fp(R\$)	P.Multa ponta(R\$)	P. multa f. ponta(R\$)	Total Azul(R\$)
1	3971	2555	82,36	2067	0	0	8675,36
2	3971	2555	102,95	4134	0	0	10762,95
3	3971	2555	82,36	3100,5	0	0	9708,86
4	3971	2759,4	205,9	2687,1	0	722	10345,4
5	3971	2759,4	82,36	4134	0	722	11668,76
6	5415	2708,3	308,85	2067	1444	541,5	12484,65
7	3971	2555	154,425	6201	0	0	12881,425
8	5595,5	2555	82,36	2480,4	1624,5	0	12337,76
9	3971	2555	82,36	2687,1	0	0	9295,46
10	3971	2555	411,8	2893,8	0	0	9831,6
11	4512,5	2555	82,36	3513,9	541,5	0	11205,26
12	5415	2555	82,36	2067	1444	0	11563,36

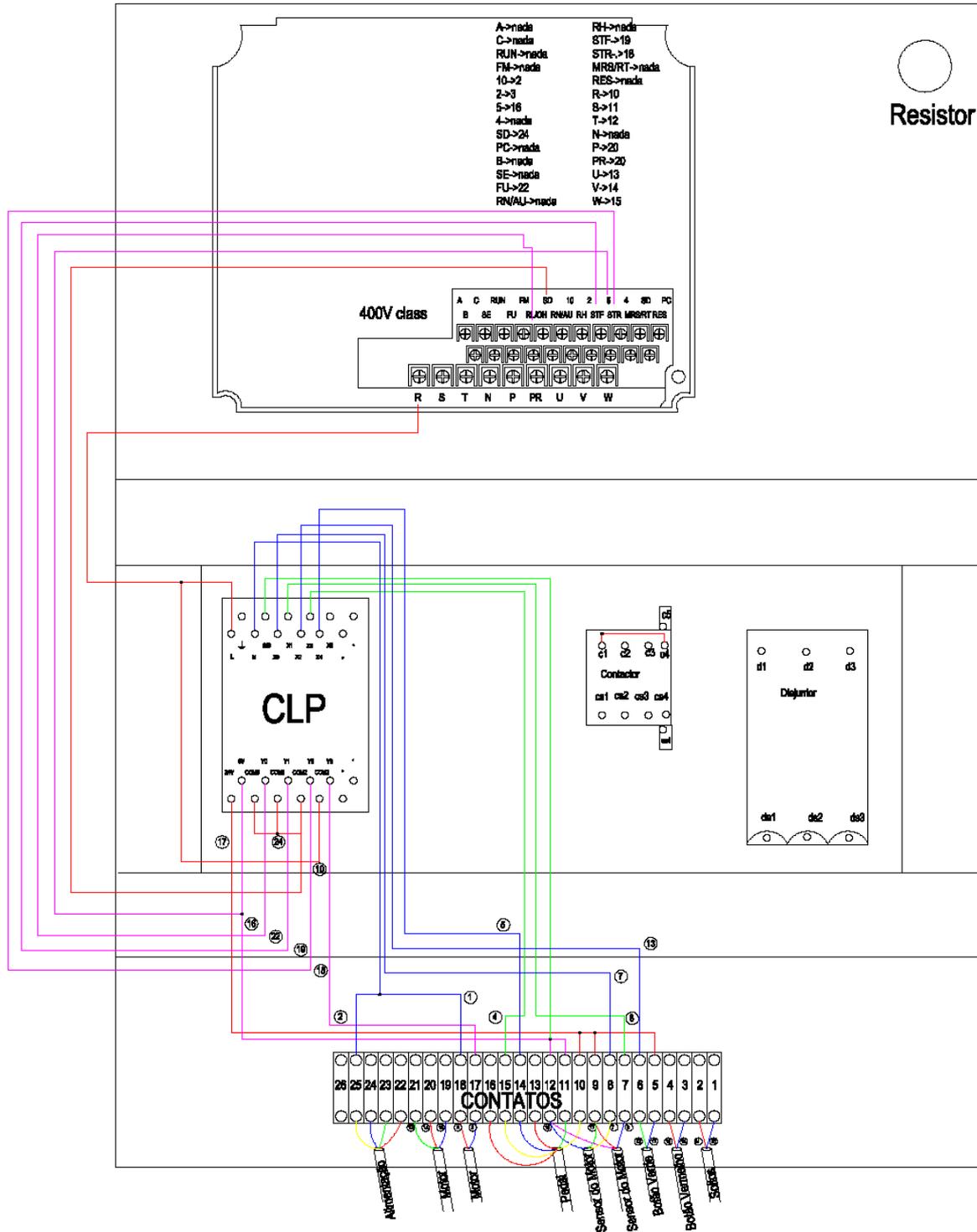
Tabela da Tarifa Branca

Tarifa Branca				
Mês	P.consumo ponta(R\$)	P.consumo f.ponta(R\$)	P.Consumo intermediário(R\$)	Total
1	164,72	1029,5	54,04875	124
2	205,9	2059	54,04875	231
3	164,72	1544,25	54,04875	176
4	411,8	1338,35	54,04875	180
5	164,72	2059	54,04875	227
6	617,7	1029,5	54,04875	170
7	308,85	3088,5	92,655	34
8	164,72	1235,4	54,04875	145
9	164,72	1338,35	54,04875	155
10	823,6	1441,3	77,2125	23
11	164,72	1750,15	61,77	19
12	164,72	1029,5	54,04875	124

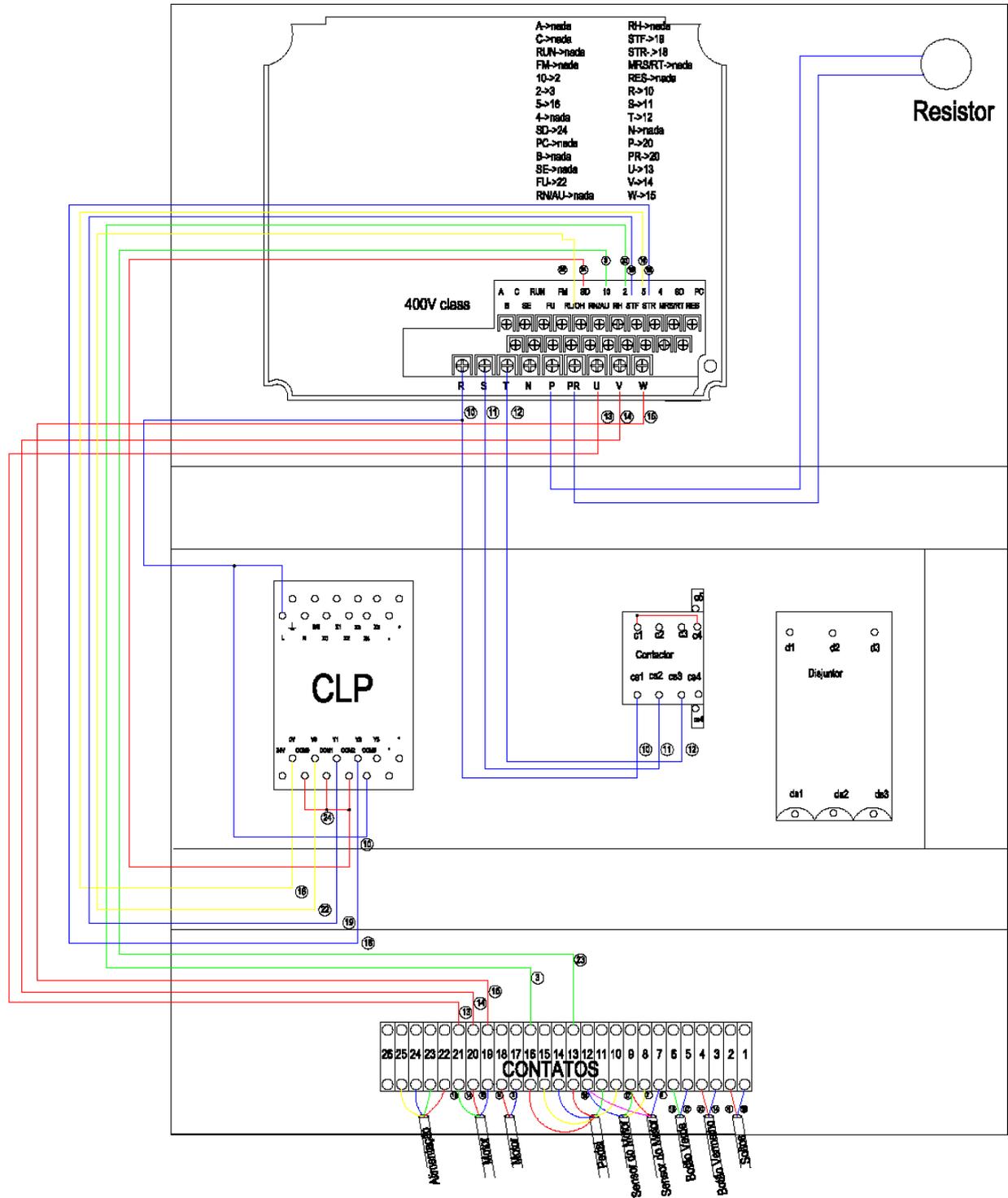
APÊNDICE F-DIAGRAMAS DA BOBINADEIRA.

Aqui estão os diagramas da bobinadeira. Foi feito um diagrama para cada componente para facilitar a análise. Por fim foi feito o desenho completo, que deveria ser colocada em papelA2

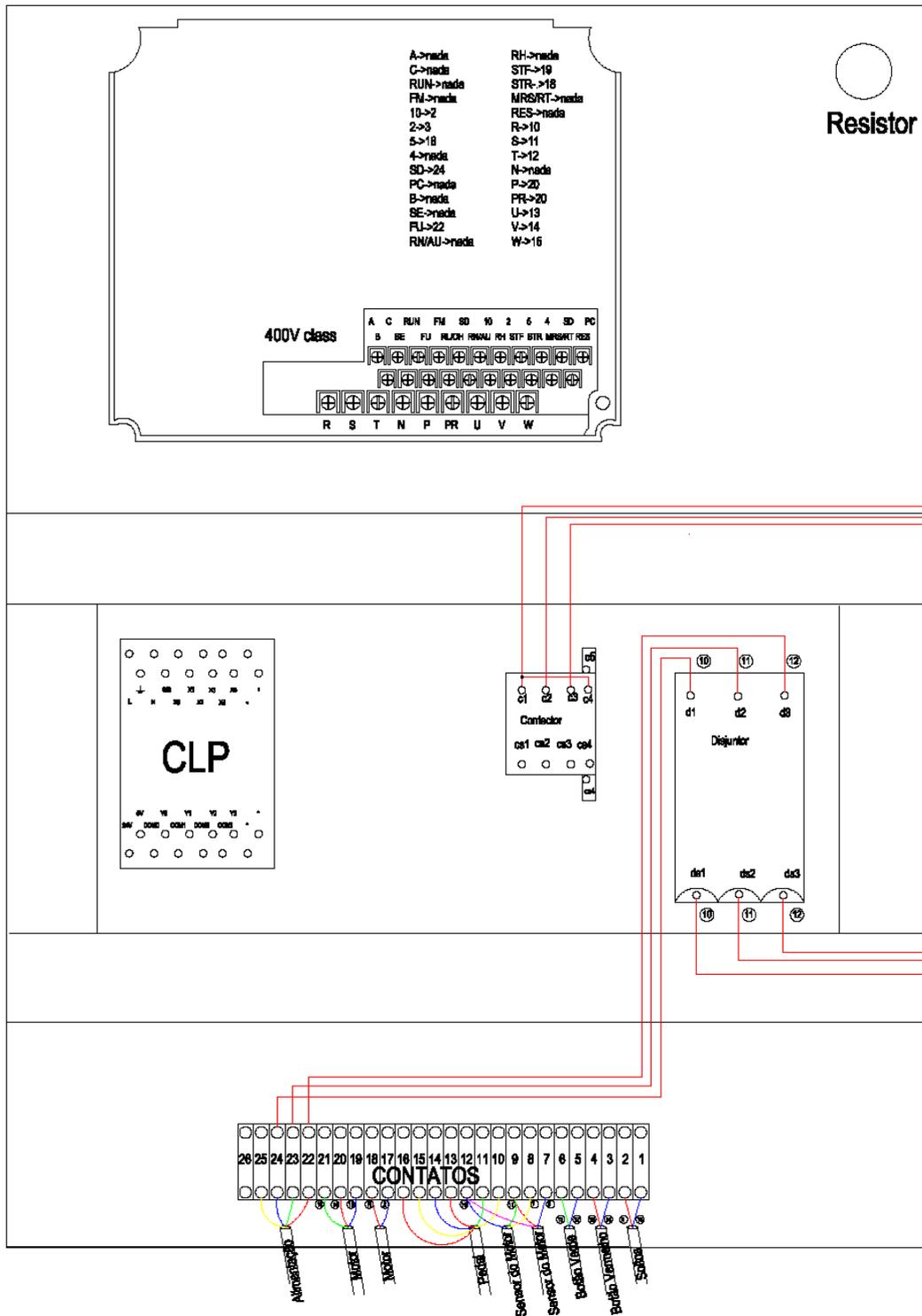
Ligações do CLP



Ligações do inversor



Ligações do disjuntor



Legenda de

1-NO
2-NO
3-NO
4-NO
5-NO
6-NO
7-NO
8-NO
9-NO
10-NO
11-NO
12-NO
13-NO
14-NO
15-NO
16-NO
17-NO
18-NO
19-NO
20-NO
21-NO
22-NO
23-NO
24-NO

