

**INSTITUTO FEDERAL DE MINAS GERAIS
BACHARELADO EM ENGENHARIA ELÉTRICA
CELSO DE OLIVEIRA PEREIRA JÚNIOR**

**PROTÓTIPO DE SISTEMA EMBARCADO PARA MONITORAMENTO E AUXÍLIO
NA ADEQUAÇÃO DE PERFIL DE CONSUMO DE ENERGIA À TARIFA BRANCA**

FORMIGA - MG

2022

CELSO DE OLIVEIRA PEREIRA JÚNIOR

**PROTÓTIPO DE SISTEMA EMBARCADO PARA MONITORAMENTO E AUXÍLIO
NA ADEQUAÇÃO DE PERFIL DE CONSUMO DE ENERGIA À TARIFA BRANCA**

Trabalho Conclusão de Curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Minas Gerais, Campus Formiga, como requisito para obtenção do título de Bacharel em Engenharia Elétrica.

Área de Concentração: Eletrônica e Sistemas Embarcados

Orientador: Prof. Dr. Lucas Frederico Jardim Meloni
Coorientador: Prof. Ms. Marco Antônio Silva Pereira

FORMIGA - MG

2022

Ficha de identificação da obra

A large, empty rectangular box with a thin black border, intended for entering identification data for a work.

“Ideias não surgem totalmente formadas. Elas só se tornam claras à medida que você trabalha nelas. Você apenas tem que começar.”

Mark Zuckerberg

AGRADECIMENTOS

Dou início agradecendo primeiro a Deus, por permitir que passasse por todos esses anos de graduação com saúde e em condições para seguir com meus estudos.

Agradeço também a minha família por sempre me apoiar e por terem me dado todo suporte necessário. Em especial agradeço a minha mãe Nilza, que esteve durante todos esses anos sempre me aconselhando e me dando ânimo para não desistir.

Agradeço ao meu orientador professor Dr. Lucas Frederico Jardim Meloni e ao coorientador professor Ms. Marco Antônio Silva Pereira por toda orientação, compreensão e paciência durante o desenvolvimento deste trabalho.

A todos meus amigos, em especial aos que tive a oportunidade de dividir a maior parte do meu tempo durante a graduação, José Elias, Gregory, Joao Rafael, Lucas, Thales e Daniel.

A todos os professores e funcionários do campus IFMG Formiga que graças ao trabalho desses profissionais estou tendo a oportunidade de concluir minha graduação.

RESUMO

Grande parte dos consumidores domésticos brasileiros concentram seu consumo diário de energia elétrica na faixa horária denominada “Horário de Ponta”. Para tentar mudar este cenário e motivar um consumo melhor distribuído ao longo do dia, foi proposto e implantado, pela Agência Nacional de Energia Elétrica (ANEEL), o modelo denominado “Tarifa Branca”, que prevê a cobrança de valores diferentes para a energia elétrica, dependendo o horário do dia. Comparado com o modelo tarifário convencional, apesar do valor da “Tarifa Branca” ser menor fora de ponta, as tarifas para os horários próximos ou dentro do “Horário de Ponta” são consideravelmente maiores. Logo, os consumidores que aderirem a este modelo, mas não modificarem seus perfis de consumo adequadamente, serão passíveis de gastar mais em suas contas de luz. Assim, com intuito de auxiliar os consumidores a adaptarem-se a “Tarifa Branca”, propõe-se nesse trabalho um sistema embarcado capaz de registrar e atuar emitindo alertas, evitando que o consumo diário se concentre dentro do “Horário de Ponta”. Para monitorar o consumo foram utilizados sensores de corrente e tensão, bem como um relógio de tempo real (RTC) para auxiliar no registro da data e o horário das medições. Também foi utilizado um cartão de memória externo, que armazenará os dados coletados, possibilitando suas análises através de *softwares* editores de planilhas dinâmicas. Todos estes componentes foram controlados e supervisionados por um microcontrolador, cujas etapas de estruturação do *firmware* e também de configuração do sistema *hardware* serão abordadas. Por fim, verificou-se que o sistema proposto é capaz de auxiliar os consumidores a concentrarem seus consumos fora do “Horário de Ponta” evitando prejuízos ao aderirem a “Tarifa Branca”.

Palavras-chave: Tarifa Branca; Perfil de consumo de energia; Sistemas Embarcados.

ABSTRACT

A large part of Brazilian domestic consumers concentrate their daily consumption of electricity in the time slot called "Peak Time". To try to change this scenario and motivate a better distribution of consumption throughout the day, the National Electric Energy Agency (ANEEL) proposed and implemented the model called "White Tariff", which provides for charging different amounts for electric energy, depending on the time of day. Compared with the conventional tariff model, despite the value of the "White Tariff" being lower off-peak, the tariffs for hours close to or within the "Peak Hours" are considerably higher. Therefore, consumers who adhere to this model, but do not modify their consumption profiles properly, will be likely to spend more on their electricity bills. Thus, in order to help consumers adapt to the "White Tariff", this work proposes an embedded system capable of recording and acting by issuing alerts, preventing daily consumption from being concentrated within the "Peak Time". Current and voltage sensors were used to monitor consumption, as well as a real-time clock (RTC) to assist in recording the date and time of measurements. An external memory card was also used, which will store the collected data, enabling their analysis through dynamic spreadsheet editor software. All these components were controlled and supervised by a microcontroller, whose firmware structuring and hardware system configuration steps will be addressed. Finally, it was found that the proposed system is capable of helping consumers to concentrate their consumption outside the "Peak Hours" avoiding losses when adhering to the "White Rate".

Keywords: "Tarifa Branca", Energy consumption profile, Embedded systems.

LISTA DE FIGURAS

Figura 1- Matriz energética brasileira.	15
Figura 2 – Consumo por classe (GWh).	17
Figura 3 – Linhas de Transmissão do Sistema Interligado Nacional.	18
Figura 4 – Faixas de horários da tarifa branca.	20
Figura 5 – Sensor de tensão ZMPT101B	31
Figura 6 - Conexão do módulo ZMPT101B a placa Arduino UNO.	28
Figura 7 – Sensor de corrente SCT-03	29
Figura 8 – Circuito utilizado para condicionar o sinal de saída do sensor de corrente	30
Figura 9 - Circuito de condicionamento do SCT-013-000.....	21
Figura 10 – Módulo RTC Ds1302.....	22
Figura 11 – Conexão do módulo SD	22
Figura 12- Fluxograma de operação do sistema embarcado	25
Figura 13 – Fluxograma para descrição do algoritmo de monitoramento e tomada de decisão	25
Figura 14 – Obtenção do consumo médio diário através de uma fatura de conta de energia	26
Figura 15 – Montagem sistema completo	27
Figura 16 – Instalação do protótipo em um quadro de distribuição monofásico.	31
Figura 17 – Bibliotecas utilizadas	33
Figura 18 – Método utilizado para calcular o consumo de energia através da potência instantânea.	34
Figura 19 – Tela de ajuste do horário.....	35
Figura 20 – Menu de Monitoramento em estado de entrada.....	35
Figura 21 - Menu de monitoramento em estado de exibição de grandezas.	35
Figura 22 – Menu de entrada do valor de consumo máximo diário.....	36
Figura 23 – Exibição do consumo diário máximo recomendado.....	36
Figura 24 - Menu data e hora.	36
Figura 25 – Exibição de data e hora no menu.	37
Figura 26 – Página para importação dos dados salvos em arquivo texto no cartão SD.	38
Figura 27 – Gráfico do perfil de consumo diário obtido através dos registros.	39

Figura 28 – Página “Perfil de consumo”	40
Figura 29 – Análise do perfil de consumo diário registrado pelo sistema embarcado.	41
Figura 30 – Protótipo do sistema embarcado para monitoramento e auxílio na adequação de perfil de consumo de energia à tarifa branca	42
Figura 31 – Layout da placa confeccionada com auxílio do <i>Software</i> Labcenter Proteus 8.5.	42
Figura 32 – Vista tridimensional da placa desenvolvida no <i>Software</i> Labcenter Proteus 8.5.	43
Figura 33 – Imagem do segundo protótipo.	44
Figura 34 – Análise do consumo concentrado horário de pico.	45
Figura 35 – Análise do resultado do consumo concentrado no período intermediário.	46
Figura 36 – Análise do consumo concentrado no período fora de ponta.	47
Figura 37 – Análise de resultados distribuídos ao longo de um dia.	47

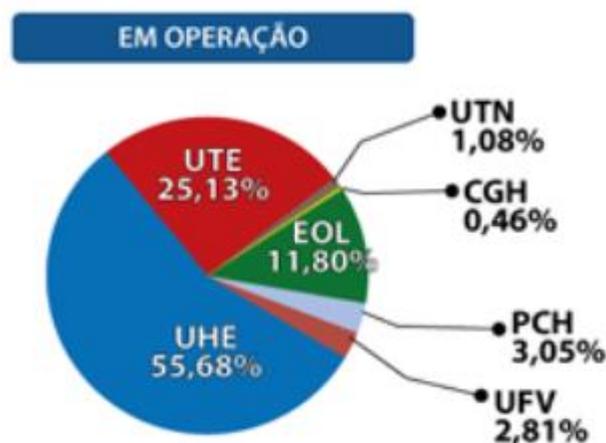
SUMÁRIO

1	INTRODUÇÃO	15
1.1	JUSTIFICATIVA	17
1.2	OBJETIVOS	17
2	FUNDAMENTAÇÃO TEÓRICA	18
3	DESENVOLVIMENTO.....	21
3.1	DISPOSITIVOS UTILIZADOS.....	21
3.2	PROPOSTA DO SISTEMA.....	27
3.3	RELAÇÃO DE EQUIPAMENTOS E COMPONENTES INVESTIDOS NO PROTÓTIPO	32
3.4	DESENVOLVIMENTO DO <i>SOFTWARES</i>	32
3.4.1	LÓGICA DE FUNCIONAMENTO	34
3.4.2	TRATAMENTO DE DADOS.....	37
4	RESULTADOS	41
5	CONCLUSÃO.....	48
	REFERÊNCIAS.....	50

1 INTRODUÇÃO

A matriz energética brasileira é caracterizada por apresentar grande dependência de recursos hídricos, como ilustrado no gráfico mostrado na Figura 1, o qual foi retirado da notícia divulgada pela ANEEL por meio do site do Governo federal (ANEEL,2022). Mais de 60% da energia elétrica nacional é produzida através de usinas hidrelétricas. Essa dependência é mais visível em períodos de secas prolongadas, quando entram em vigência as bandeiras tarifárias amarela, vermelhas e de escassez hídrica, quando são cobradas taxas extras para cada centena de kWh consumidos (CEMIG, 2021). Além disso, também são incidentes demais taxas e encargos, bem como reajustes tarifários que também contribuem para o aumento da tarifa energética perante a inflação (CORSINI, 2022).

Figura 1- Matriz energética brasileira.



Fonte: Imagem retirada de (ANEEL, 2022).

Mesmo sendo passível de gastar mais em períodos de estiagem, o consumidor brasileiro ainda possui alternativas para economizar na sua conta de energia elétrica. Um exemplo, cujos benefícios são vistos à longo prazo, é aderir a instalação de sistemas fotovoltaicos para produção de energia (CAMARGO, 2017). Nestes sistemas de geração regulamentado pela Resolução Normativa ANEEL nº 482/2012 e pela Lei nº 14.300/2022, o consumidor poderá instalar em sua residência painéis fotovoltaicos e produzir sua própria energia, sendo também capaz de fornecer o excedente para a rede de distribuição de sua localidade (ELÉTRICA, 2022a). Entretanto, tal alternativa requer um investimento inicial para aquisição e instalação de equipamentos.

Outra forma é através da adesão de modelos tarifários diferentes, como a “Tarifa Branca” criada pela ANEEL (Agência Nacional de Energia Elétrica), que se tornou válida a partir de 1º de Janeiro de 2018 (ABRADEE, 2021). Trata-se de uma opção de faturamento que possibilita descontos para usuários que concentrarem seu consumo fora do “Horário de Ponta”, compreendido entre dezoito e vinte e uma horas (ELÉTRICA, 2022b). Para isso é necessário que o consumidor tenha certos cuidados, para que sua conta não fique mais cara do que se aderisse a Tarifa Convencional, na qual um mesmo valor é cobrado para toda a energia consumida ao longo do dia.

Neste contexto, algumas ferramentas podem ser utilizadas para auxiliar o usuário no monitoramento de seu consumo diário. Um exemplo são os medidores de energia inteligentes (*Smart meters*), capazes de disponibilizar informações detalhadas sobre o consumo, possibilitar a produção de relatórios para avaliação e controle de gastos, além de informar sobre ocorrência de faltas como curtos-circuitos ou interrupções momentâneas de fornecimento (TONELLI, 2019). Há também esforços para o desenvolvimento de medidores de energia cognitivos, capazes de auxiliar os consumidores a monitorarem seus perfis de consumo global ou de circuitos individuais, emitindo mensagens em interfaces simples e legíveis (BITTENCOURT; MARTINS; GASTALDELLO, 2021). Entretanto, os medidores apenas monitoram e registram dados sobre o consumo, não atuando em tomadas de decisões para melhorar ou modificar o perfil de consumo dos clientes.

Este trabalho aborda o desenvolvimento de um sistema embarcado capaz de supervisionar o consumo de energia diário e desligar algum circuito ou disparar alarmes, caso o mesmo concentre-se dentro ou nas faixas intermediárias do “Horário de Ponta”, evitando que o usuário tome prejuízos ao aderir o modelo da Tarifa Branca. Através de dados de consumo armazenados em um cartão de memória externo, bem como auxílio do uso de *softwares* de edição de planilhas dinâmicas, também será possível produzir relatórios que auxiliarão o usuário a adequar-se a tarifa branca, sugerindo como o mesmo poderá modificar seus horários de maior consumo.

1.1 JUSTIFICATIVA

A automação residencial, vem se tornando cada vez mais popular, por proporcionar tanto um conforto maior, quanto otimização de tempo e praticidade para o usuário (PEREIRA et al., 2020). Junto a isso, encontramos um cenário onde as tarifas energéticas no Brasil aumentam cada vez mais, seja por aumento da inflação, períodos de escassez hídrica ou modificações em encargos. Desse modo, encontramos um aumento na demanda energética residencial e rural, como mostrado na Figura 2, relatada no Balanço Energético Nacional feito pela Empresa de Pesquisa de Energia em 2021.

Figura 2 – Consumo por classe (GWh).

	2016	2017	2018	2019	2020	$\Delta\%$ (2020/2019)	Part. % (2020)	
Brasil	461.780	467.161	474.820	482.226	475.648	-1,4	100,0	Brazil
Residencial	132.872	134.369	137.615	142.781	148.173	3,8	31,2	Residential
Industrial	165.314	167.398	169.625	167.684	166.335	-0,8	35,0	Industrial
Comercial	87.873	88.292	88.631	92.075	82.522	-10,4	17,3	Commercial
Rural	27.267	28.136	29.168	28.870	30.908	7,1	6,5	Rural
Poder público	15.096	15.052	15.076	15.752	12.764	-19,0	2,7	Public Sector
Iluminação pública	15.035	15.443	15.690	15.850	15.463	-2,4	3,3	Public Lighting
Serviço público	14.969	15.196	15.778	15.958	16.345	2,4	3,4	Public Service
Próprio	3.355	3.277	3.238	3.257	3.138	-3,7	0,7	Own Use

Fonte: imagem retirada de (ENERGÉTICA, 2021).

Diante deste cenário, é válido qualquer esforço para poupar energia, ou então diminuir seus respectivos gastos. A principal justificativa para este trabalho é propor um sistema embarcado capaz de auxiliar o usuário a adaptar seu perfil de consumo à tarifa branca, evitando que o mesmo tenha prejuízos financeiros.

1.2 OBJETIVOS

O objetivo geral deste trabalho é auxiliar o usuário a modificar seu perfil de consumo de energia elétrica, para que o mesmo se adeque ao modelo da “Tarifa Branca” sem tomar prejuízos. O projeto também registrará informações de consumo as quais serão analisadas para produzir relatórios que possam auxiliar e instruir o usuário neste processo. Por fim, em uma

atitude extrema o sistema permitirá o chaveamento de circuitos, com o auxílio de módulo relé, desligando-os caso o usuário habilite esta função.

2 FUNDAMENTAÇÃO TEÓRICA

Segundo (SICARI, 2020), a partir da década de 1990 foi considerado o período onde iniciaram-se as privatizações, e a promulgação de leis que atrelavam reajustes tarifários em função dos custos de produção, bem como o Programa Nacional de Desestatização (PND) que tornou independentes as áreas de geração, comercialização, distribuição e transmissão. Segundo (ALMEIDA, 2008), este foi um dos grandes marcos responsáveis pelo processo de privatizações no país permitindo a entrada de capital e investimentos no setor elétrico.

O sistema elétrico brasileiro é hidro-termo-eólico, sendo a maior parcela da produção de energia gerada por usinas hidrelétricas, com vários proprietários e localizadas longes dos grandes centros, também sendo necessária a conexão e transmissão desta aos consumidores. Tal integração é denominada Sistema Interligado Nacional (SIN), sendo dividida nas regiões norte, nordeste, centro-oeste, sul e sudoeste. O SIN é responsável por entregar cerca de 96,6% de toda energia gerada (SICARI, 2020), como ilustrado na Figura 3.

Figura 3 – Linhas de Transmissão do Sistema Interligado Nacional.



Fonte: imagem retirada de (ONS, 2022).

A grande dependência de recursos hídricos e baixa diversidade energética em sua matriz, já fez com que em 2001 o Brasil fosse atingido por uma forte política de racionamento de energia, denominada “apagão” (ANDRADE, 2021). Anos depois, apesar do país ter passado por uma série de investimentos no setor energético, ainda há uma grande dependência de

recursos hídricos (ENERGÉTICA, 2020a), trazendo à tona o receio de novos episódios de “apagões”. A partir de 1º de Janeiro 2015, durante um novo período de estiagem, entra em vigor o sistema de bandeiras tarifárias e de escassez hídrica (SICARI, 2020).

Neste modelo, o custo para a produção de energia elétrica é atualizado de uma forma mensal, dependendo do nível dos reservatórios e da produção de energia através de usinas térmicas. Segundo (CPFL, 2021), com as bandeiras há a sinalização do custo mensal de geração que será cobrado ao consumidor, permitindo que o mesmo seja capaz de adaptar-se e modificar seu perfil.

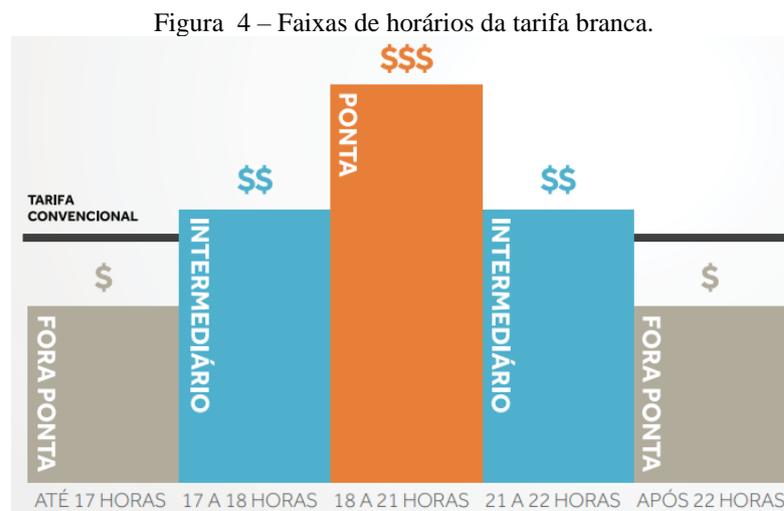
Entretanto, diferente do que ocorria em 2001, o Brasil passa por uma tendência em diversificar sua produção energética através da utilização de fontes renováveis como os sistemas fotovoltaicos, principalmente em estados onde a tarifa ainda é maior, devido a impostos (RENOVÁVEIS, 2021).

A tarifa branca, instituída pela ANEEL a partir de 1º de Janeiro de 2018 trata-se de uma modalidade tarifária diferenciada, na qual o valor pago dependerá da faixa horária em que ocorrerá o consumo de energia (MALAR, 2021). Esta modalidade não está disponível para unidades consumidores da subclasse baixa renda, iluminação pública e unidades que contam com sistema de pré-pagamento, sendo disponível para consumidores do tipo B em baixa tensão (ABRADEE, 2021). Segundo (ROSELLI et al., 2021) o objetivo da tarifa branca é aumentar a eficiência do uso das redes de distribuição e incentivar a mudança do consumo marginal do período de ponta para o período fora de ponta. Esse incentivo é interessante pelo fato de que é nos horários de ponta e intermediários que normalmente a maioria das pessoas estão retornando para suas casas, utilizando chuveiros, e demais equipamentos elétricos, o que gera um pico muito grande de consumo, exigindo que outras formas de geração de energia sejam utilizadas, como por exemplo termoeletricas que tem um custo de geração mais elevado, encarecendo o preço do kWh nos horários de ponta.

O preço cobrado pela conta de luz, é em grande parte, formado por encargos e tributos. Segundo (IDEC, 2021), cerca de 16% do custo são encargos e 28% tributos, totalizando cerca de 44% do valor total. Os encargos podem ser definidos, segundo (ENERGIA, 2021) como custos suportados pelas concessionárias e que são instituídos por leis, sendo repassados aos consumidores, como medidas para manter o equilíbrio econômico-financeiro contratual entre estas empresas e o Estado. Já os tributos, segundo (IDEC, 2021), são valores pagos diretamente ao governo, de modo a garantir o desenvolvimento de atividades do poder público. Estão inclusos os impostos de programas de integração social (PIS), contribuição para o

financiamento da seguridade social (Cofins), o custeio do serviço de iluminação pública (CIP) e o imposto sobre a circulação de mercadorias e serviços (ICMS).

Assim, a “Tarifa Branca” não isenta o consumidor de passar pelas bandeiras tarifárias ou de sofrer aumentos em sua conta de luz, mas propõe um incentivo para que o mesmo modifique seu perfil de consumo. A cobrança de valores diferentes dependerá do horário em que a eletricidade é utilizada. Segundo a ANEEL, para que o usuário seja beneficiado neste modelo é essencial que seu maior consumo esteja localizado fora dos horários de ponta ou intermediários. Para isso, o usuário deverá realizar um estudo prévio das faturas de energia anteriores assim como o dia a dia de consumo de sua residência. Dentro do modelo tarifário é possível observar 3 tipos de valores classificados em faixas fora de ponta, intermediária e ponta, como mostrado na Figura 4 (ABRADEE, 2021). Em feriados nacionais e finais de semana, o valor de todas as horas é considerado fora de ponta. Todos os períodos são homologados pela ANEEL nas revisões tarifárias periódicas de cada concessionária.



Fonte: imagem retirada de (ABRADEE, 2021).

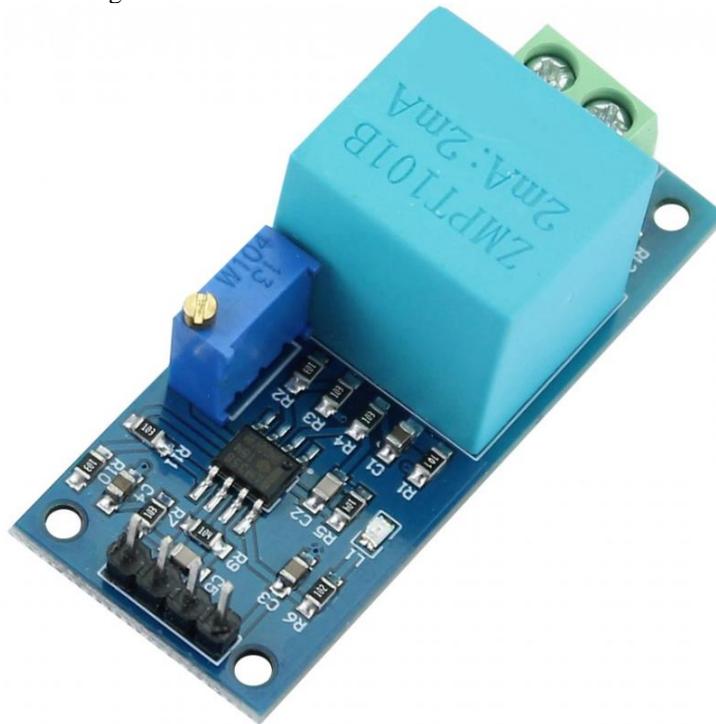
Pelo esquema simplificado mostrado na Figura 4 verifica-se que neste modelo tarifário é possível obter um custo maior na conta de luz, caso o consumidor concentre seu gasto principalmente nos horários intermediário e de ponta. Logo, cabe a concessionária monitorar os horários de consumo e também aplicar as tarifas. Assim, uma forma de auxiliar o consumidor a conseguir adequar-se a este modelo tarifário é supervisionar seu consumo ao longo do dia e indicar quais momentos o mesmo se intensifica. Caso esses momentos estejam localizados dentro dos horários intermediários ou de ponta, gera-se um alerta e uma recomendação para trocar de modelo tarifário, evitando que o mesmo tome prejuízos.

3 DESENVOLVIMENTO

3.1 DISPOSITIVOS UTILIZADOS

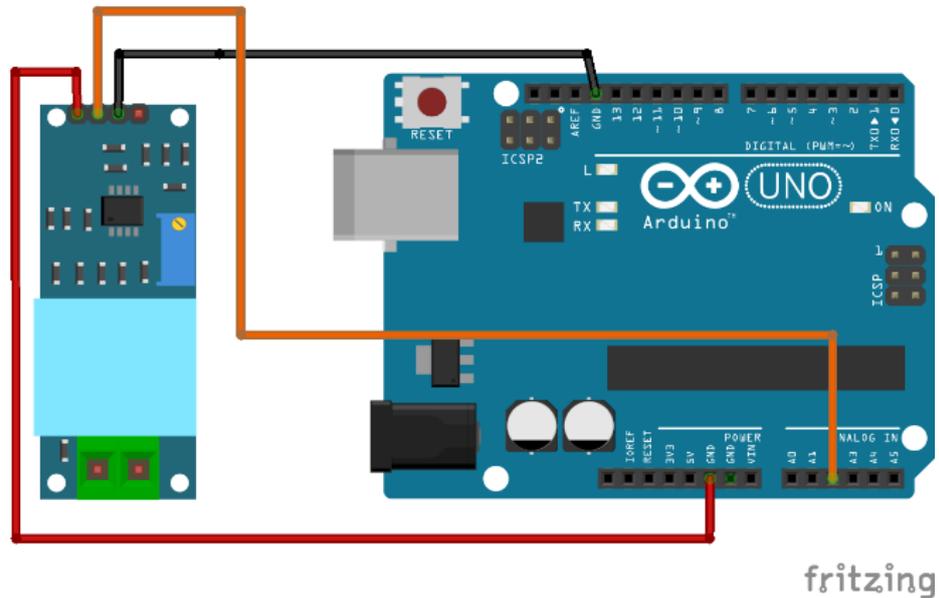
Para a leitura dos valores de tensão utilizou-se o sensor de modelo ZMPT101b, mostrado na Figura 5 e sua respectiva conexão a placa Arduino mostrada na Figura 6. Este sensor atua como um pequeno transformador de potencial, cujo primário é conectado ao lado de tensão alternada de 0 a 250Vrms e produz em seu secundário uma tensão menor que será lida pela placa Arduino para realizar a conversão Analógico para Digital (ADC). Para seu funcionamento é necessário alimentar o módulo com tensão contínua de 5 a 30V, através dos pinos GND e VCC. Além disso o sensor apresenta duas portas de entradas (bornes) onde é realizado a leitura do sinal amostrado, as quais são ligadas em paralelo ao cabo fase e neutro do padrão de entrada. Sua temperatura de operação vai de -40° a 70°C com precisão de leitura de $\pm 1\%$ (INFO, 2022).

Figura 5 - Sensor de tensão ZMPT101B.



Fonte: imagem retirada de (INFO, 2022).

Figura 6– Conexão do módulo ZMPT101B a placa Arduino UNO.



Fonte: Imagem elaborada pelo próprio autor com auxílio do *software* Fritzing (2022).

Para o perfeito funcionamento do sensor é necessário calibrar o módulo dispositivo. Isso se deve ao fato de que o módulo pode vir de fábrica produzindo uma forma de onda cortada no semiciclo positivo, tendendo a formar uma onda quadrada, o que pode comprometer a leitura feita pelo pino analógico do Arduino. Para isso, é necessário ligar as entradas de energia a uma tomada, alimentar o sensor e conectar a sua saída de sinal a um osciloscópio e por meio do potenciômetro já embutido no circuito do sensor fazer o ajuste da forma de onda para que a mesma fique o mais próximo possível de uma senoide (OLIVEIRA, 2018).

Para monitorar a corrente foi utilizado o sensor SCT-03, mostrado na Figura 7. Trata-se de um *Split-core Current Transformer* (SCT), ou seja, transformador de corrente de núcleo dividido, utiliza uma bobina interna, onde por meio do princípio da Lei de Faraday, o campo eletromagnético gerado pelo fio condutor induz uma força eletromotriz que gera uma corrente na bobina. Sua especificação técnica é de 100A no primário, com saída de 50mA no secundário, possuindo um fator K de 2000 (referente à relação de espiras do Trafo) e suportando temperaturas que variam entre -25°C até 70°C . Sua faixa de operação é não linear em 3% e tem uma taxa de variação de seu valor nominal que vai de 10% a 120% equivalente a 10A até 120A (DE BRITO, 2016). Este sensor é prático por não ser necessário contato com o circuito elétrico para a medição de corrente, e seu funcionamento é análogo ao de um alicate amperímetro. O fio forma o enrolamento primário do transformador. O núcleo do transformador de ferro concentra o campo e o acopla ao enrolamento secundário. e, desde que o campo magnético

mude continuamente, faz com que uma corrente flua também nesse enrolamento. Essa corrente deve fluir em um resistor de carga, o que gera uma tensão que o Arduino consegue identificar. É importante destacar que um transformador de corrente não mede corrente contínua (DE BRITO, 2016).

Figura 7 – Sensor de corrente SCT-03.



Fonte: Do próprio autor (2022).

Devido ao Arduino conseguir efetuar leitura de níveis de tensão entre 0 e 5V, para que o sensor funcione corretamente é necessário realizar ajustes de hardware, adicionando resistores e capacitores ao circuito. Na saída do sensor é gerada uma variação de corrente, que será detectada pelo após a inclusão de um resistor de carga também chamado de “Burden Resistor” (DE BRITO, 2016). Ainda segundo (DE BRITO, 2016) o valor deste resistor pode ser determinado através da expressão (1).

$$I (\text{medido}) = \sqrt{2} * I_{\text{max}} (\text{rms}) = 1,414 * 100A = 141,4 A \quad (1)$$

Em seguida é necessário encontrar o número total de espiras no secundário, através da expressão (2), onde I_1 é a corrente eficaz no primário, a qual será monitorada, I_2 é a corrente produzida no secundário do sensor e k é a relação de transformação.

$$k = I_2 / I_1 \quad (2)$$

Utilizando a referência de (DE BRITO, 2016), devido a corrente de entrada do sensor poder alcançar 100 A (RMS) e sua saída até 50mA (datasheet do dispositivo), portanto: $k=100/50\text{mA} = 2000$. Com isso a corrente máxima produzida no secundário do sensor resultou

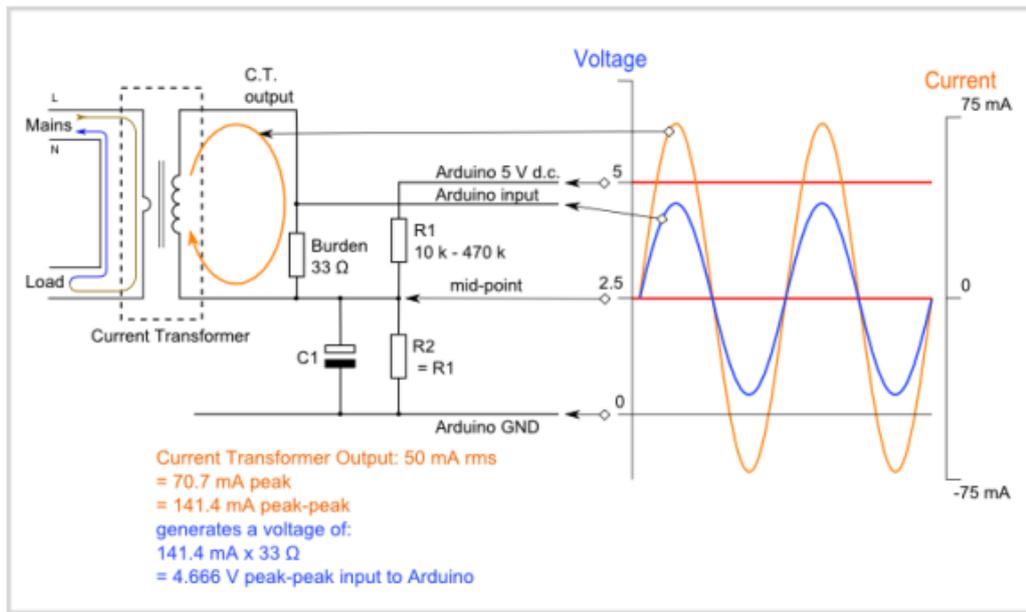
em $I_{p2} = 70,7$ mA. Para dimensionar o resistor Burden, de forma a manter uma boa resolução da leitura, utiliza-se metade do valor de tensão máximo do Arduino, conforme mostrado em (3).

$$R_{Carga} = \frac{2,5}{0,0707} = 35,4\Omega \quad (3)$$

Também foi feito um divisor de tensão, como mostrado na Figura 8, o qual serve para elevar o sinal da referência em aproximadamente 2,5 V, eliminando a tensão negativa da senoide produzida na saída do sensor. O capacitor eletrolítico é responsável por filtrar o sinal que será mandado para o AD (Analógico/ Digital) do Arduino (DE BRITO, 2016). Pelos cálculos e experimentos feitos, a entrada no Arduino será em torno de 4,5 Vpp. Definido o resistor de carga, também é necessário o cálculo do valor de calibração que posteriormente será utilizado na codificação do software.

$$V_{Calibração} = \frac{2000}{35} = 57,143 \quad (4)$$

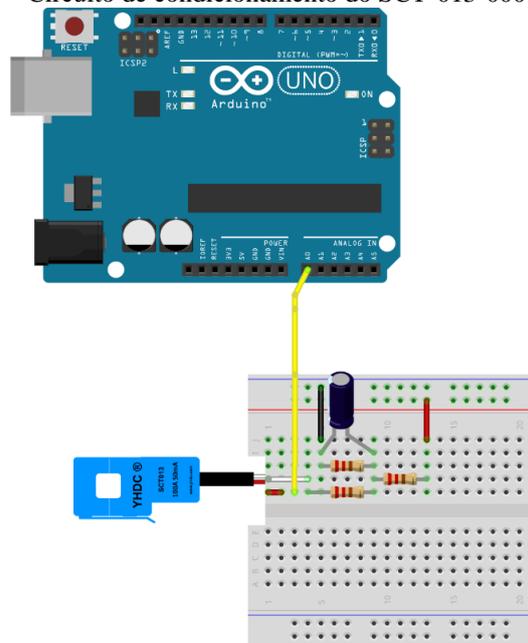
Figura 8- Circuito utilizado para condicionar o sinal de saída do sensor de corrente.



Fonte: Imagem retirada de (LEARN OPEN ENERGY MONITOR, 2021).

Após realizado todos os cálculos necessários, iniciou-se a montagem do circuito responsável por enviar as informações até o Arduino. A Figura 9 ilustra a configuração utilizada para a montagem.

Figura 9 - Circuito de condicionamento do SCT-013-000

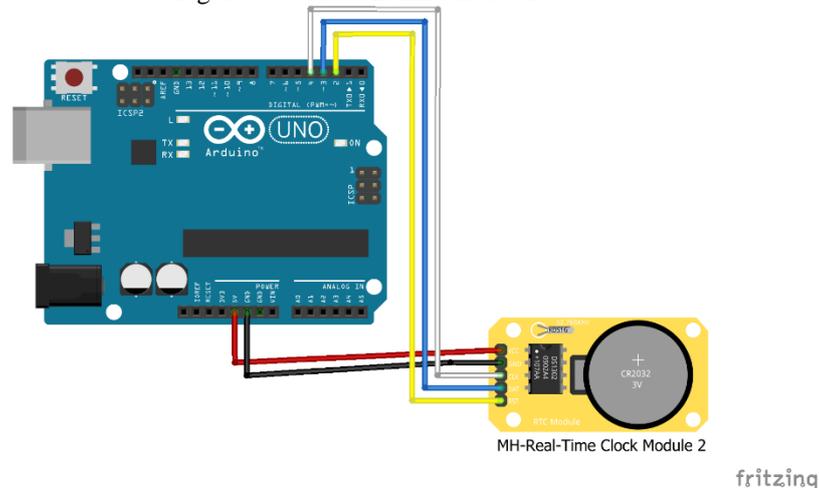


Fonte: Imagem produzida pelo próprio autor (2022)

Para o desenvolvimento do código utilizando esse sensor, na maioria dos projetos utiliza-se a biblioteca *EmonLib.h*, desenvolvida pela Energy Monitor, na linha de Open Source e sua função principal é extrair do sistema valores de fator de potência, potência ativa, potência aparente, tensão RMS e corrente RMS. Basicamente o funcionamento dessa biblioteca se baseia na contagem de ciclos que um sinal realiza em determinado intervalo de tempo, quantidade de vezes que cada sinal (corrente e tensão) cruzam o eixo para se ter um valor exato de deslocamento de fase, resultando em um sinal cada vez mais assertivo (DE BRITO, 2016).

O módulo de relógio de tempo real (RTC – Modelo Ds1301) foi escolhido para enviar os dados de data e hora para o Arduino. Sua comunicação é serial e realizada por meio dos pinos SDA e SCL, cuja conexão é mostrada na Figura 10. É capaz de fornecer as informações de data, hora, minuto e segundo, operando nos formatos 12 e 24h, além disso o módulo possui bateria, o que permite sua operação mesmo com o circuito desenergizado (VIDAL, 2017).

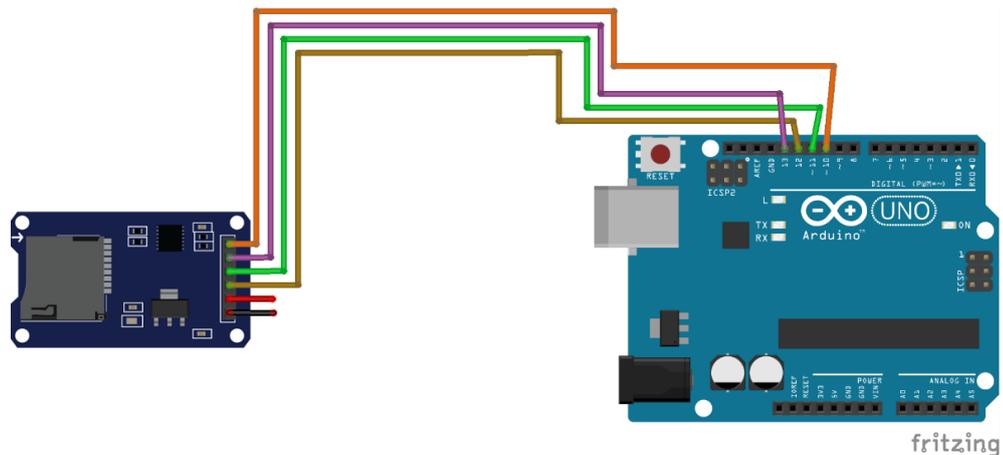
Figura 10 - Módulo RTC Ds1302



Fonte: imagem elaborada pelo próprio autor (2022)

Também foi utilizado o módulo com cartão SD, o qual permite realizar a leitura e registro de dados em arquivos simples em formato de texto. A conexão deste dispositivo é mostrada na Figura 11 e a comunicação é feita pela interface SPI, através dos pinos SCK (marcado como D13 na placa Arduino), MISO (marcado como D12 na placa Arduino), MOSI (marcado como D11 na placa Arduino) e CS (marcado como D10 na placa Arduino). O nível de sinal é de 3.3V, contudo este leitor de cartão micro SD já vem com divisor de tensão, permitindo uma ligação direta com placas que trabalham com 5V, como o Arduino.

Figura 11 – Conexão do módulo SD.



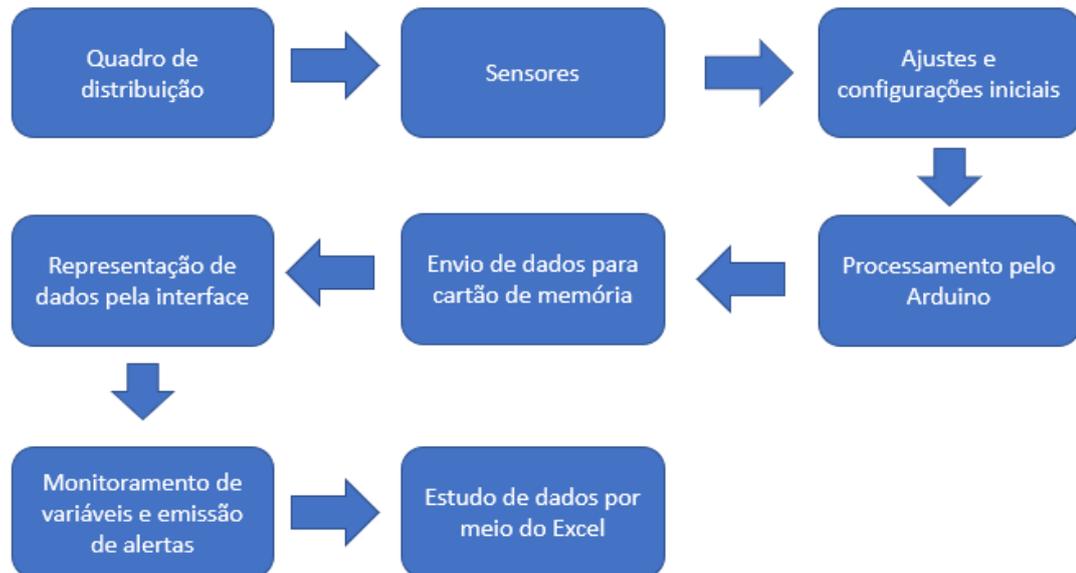
Fonte: Imagem elaborada pelo próprio autor (2022)

Além desses dispositivos citados também foi utilizado no projeto resistores, botões e display LCD com módulo I2C integrado.

3.2 PROPOSTA DO SISTEMA

O funcionamento básico do sistema embarcado de monitoramento é demonstrado na Figura 12. Inicialmente o dispositivo deve ser instalado no quadro de distribuição conforme sugere a Figura 16, juntamente dos sensores de corrente e tensão para que os mesmos possam efetuar todas as medições. Quando ligado pela primeira vez, o dispositivo exibirá telas de ajustes de data e hora para seu RTC, bem como a data de início para monitoramento do consumo de energia. Logo após, o mesmo entra em operação e indica o consumo diário em kWh, salvando estes em um cartão SD. O intervalo de aquisição e registro escolhido foi de 15 minutos, os quais são salvos através de um arquivo texto, em formato de lista. Este formato pode ser importado por *softwares* editores de planilhas, como o *Microsoft Excel*® ou o *Libre Office*. Durante sua execução, entra em funcionamento um algoritmo que monitora o consumo e se o mesmo atingir limites pré-definidos para o “Horário de Ponta”, serão emitidos alertas e acionadas as saídas a relé.

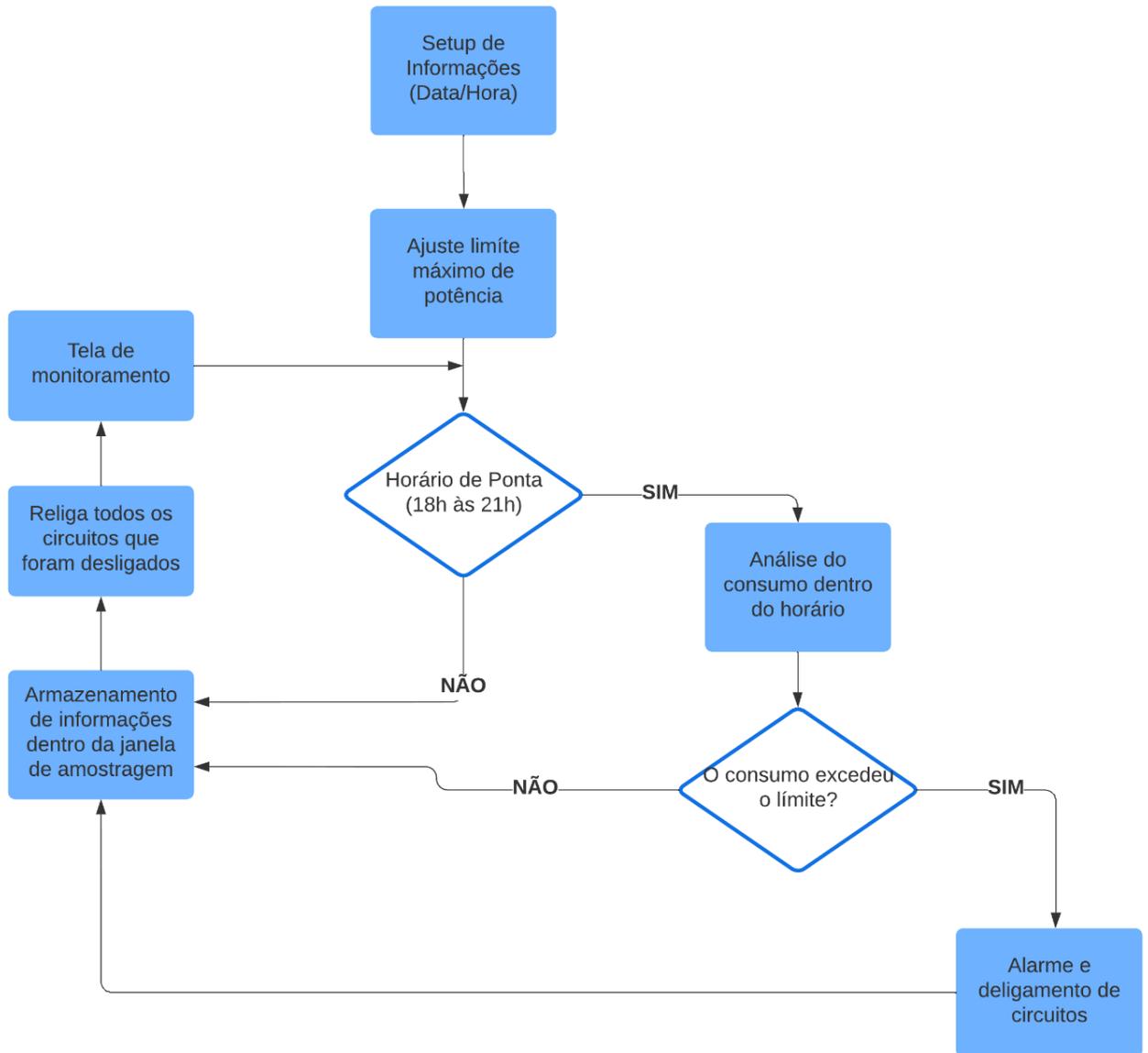
Figura 12 - Fluxograma de operação do sistema embarcado.



Fonte: Imagem produzida pelo próprio autor (2022).

O fluxograma da Figura 13 ilustra com mais detalhes o funcionamento do algoritmo. Inicialmente o sistema inicia em uma tela de *setup* onde são programadas a data e hora atuais. Em seguida, um valor diário médio de consumo é programado pelo usuário. Este valor pode ser determinado através da conta de luz, por exemplo a mostrada na Figura 14. A partir deste valor é estipulado um valor máximo de porcentagem deste consumo diário o qual não poderá ser excedido no “Horário de Ponta”, caso contrário o usuário terá prejuízos ao aderir a tarifa branca. Assim, o *software* verificará se durante o “Horário de Ponta” tal limite foi ou não ultrapassado. Caso não seja ultrapassado, o sistema retorna ao estado inicial de aquisição. Caso contrário, o mesmo emitirá um alerta ligando as saídas a relé e desativando determinados circuitos, com o intuito de alertar o usuário e evitar que o consumo se estenda.

Figura 13 – Fluxograma para descrição do algoritmo de monitoramento e tomada de decisão.



Fonte: Imagem produzida pelo próprio autor (2022).

Figura 14 – Obtenção do consumo médio diário através de uma fatura de conta de energia.

Histórico de Consumo			
MÊS/ANO	CONSUMO kWh	MEDIA kWh/Dia	Dias
JUN/22	100	3,03	33
MAI/22	81	2,89	28
ABR/22	102	3,09	33
MAR/22	90	3,00	30
FEV/22	91	3,13	29
JAN/22	94	3,24	29
DEZ/21	99	3,19	31
NOV/21	94	3,13	30
OUT/21	89	4,04	22
SET/21	0	0,00	0
AGO/21	0	0,00	0
JUL/21	0	0,00	0
JUN/21	0	0,00	0

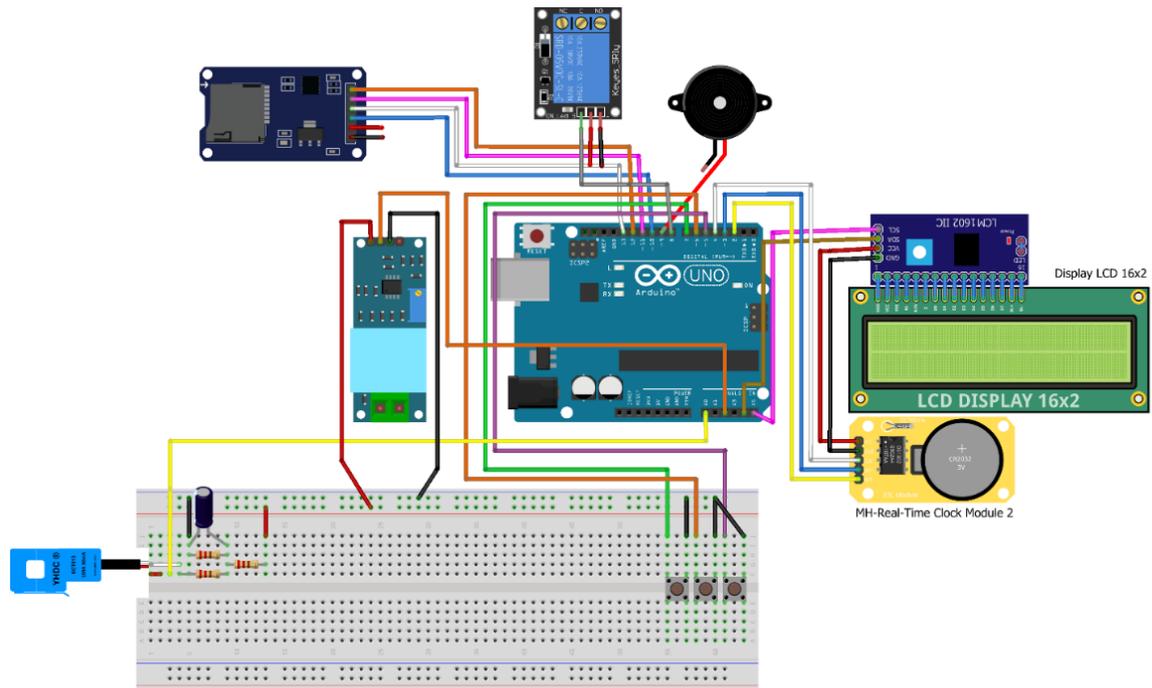
Ouvidoria CEMIG: 0800 728 3838 - Agência Nacional de Energia Elétrica - ANEEL - Tele

CEMIG

Fonte: Imagem produzida pelo próprio autor (2022).

O sistema proposto foi concebido com sua conexão realizada dentro do quadro de distribuição de circuitos (QDC) da residência, conforme ilustrado na Figura 16, que é um esquema de como o protótipo seria instalado em um padrão monofásico. Como já demonstrado na seção 3.1, para o monitoramento da corrente foi utilizado o sensor não invasivo, conectado junto ao cabo fase, na saída do disjuntor principal, com acoplamento análogo ao de um alicate amperímetro. Para a tensão, foi utilizado um transformador de potencial, cujos terminais do enrolamento primário foram ligados em paralelo aos cabos de fase e neutro do circuito. As saídas do sistema embarcado utilizarão relés que poderão atuar seccionando circuitos através de contadores. O circuito final com todos os componentes pode ser observado pela figura 15.

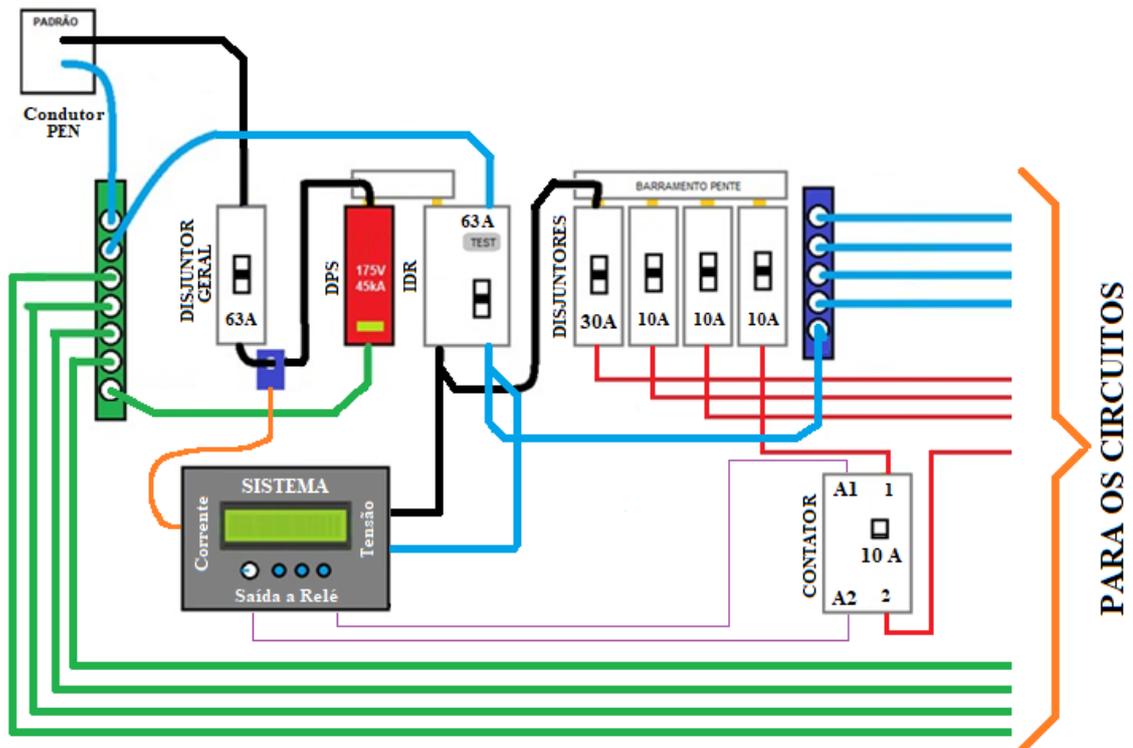
Figura 15 - Montagem sistema completo.



fritzing

Fonte: Imagem elaborada pelo próprio autor (2022)

Figura 16 - Instalação do protótipo em um quadro de distribuição monofásico.



Fonte: Imagem produzida pelo próprio autor (2022)

3.3 RELAÇÃO DE EQUIPAMENTOS E COMPONENTES INVESTIDOS NO PROTÓTIPO

Durante a realização deste trabalho, devido problemas encontrados pela pandemia causada pelo COVID-19 (PAHO, 2020) foi necessário realizar experimentos mais simples e simplificação de alguns dos circuitos mostrados anteriormente. Também não foi possível usufruir em tempo hábil dos laboratórios do IFMG *Campus* Formiga, não sendo possível utilizar osciloscópios ou demais dispositivos. De forma a reduzir custos do projeto e torna-lo mais atrativo alguns estudos foram realizados de forma a realizar testes e encontrar funcionalidades de dispositivos que atendessem ao projeto. Dessa forma, obteve-se na Tabela 1 um orçamento acessível e viável para se desenvolver o protótipo.

Tabela 1: Orçamento projeto.

Item	Qtd	Preço Unitário (R\$)	Preço Total (R\$)
Sensor corrente SCT-013 100A	1	64,40	64,40
Sensor tensão ZMPT101B	1	26,50	26,50
Capacitor	1	1,25	1,25
Resistor	5	0,10	0,50
RTC Ds1302	1	9,40	9,40
Arduino Uno	1	57,90	57,90
Push Button	3	0,30	0,90
Módulo i2C	1	10,50	10,50
Display LCD	1	20,90	20,90
Módulo cartão SD	1	9,90	9,90
Jumpers	30	0,30	9,00
TOTAL			211,15

3.4 DESENVOLVIMENTO DO SOFTWARES

O algoritmo desenvolvido nesse trabalho objetiva-se em deixar o sistema o mais dinâmico possível e assim processar as informações da melhor forma possível a fim de obter os

resultados esperados. De acordo com a Figura 17, extraída do código, é possível observar que foi necessário utilizar 7 bibliotecas.

Figura 17 – Bibliotecas utilizadas

```
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#include <DS1302.h>
#include <SPI.h>
#include <SD.h>
#include "EmonLib.h"
```

Fonte: Imagem produzida pelo próprio autor (2022).

A primeira Wire.h, permite a comunicação do Arduino com dispositivos 12C/TWI, juntamente dessa biblioteca também é utilizado a LiquidCrystal_I2C.h e a LCD.h, ambas também responsáveis pelo funcionamento do display LCD, o qual está conectado ao Arduino por meio de um módulo I2C. A biblioteca DS1302.h, é responsável pelo funcionamento do timer RTC, para a leitura e ajuste de dados de data e hora do sistema.

A biblioteca SPI.h (Serial Peripheral Interface), de grande importância para a maioria dos circuitos montados com Arduino, permite utilizar o protocolo SPI que utiliza dados seriais síncronos para comunicação entre o microcontrolador e os demais periféricos. Esta biblioteca também pode ser usada entre dois microcontroladores. A comunicação SPI do Arduino UNO possui um dispositivo "mestre" e outros dispositivos denominados como "escravos". A função do mestre consiste em realizar a coordenação de toda a comunicação, pois, ele tem a capacidade de enviar e requisitar informações aos escravos existentes na estrutura de comunicação, os quais, devem responder às requisições citadas (DE BRITO, 2016).

Para que todos os dados obtidos sejam salvos, é necessário que haja um dispositivo de memória conectado ao sistema, com isso, para a comunicação entre o Arduino e o módulo SD, utilizou-se a biblioteca SD.h.

A biblioteca EmonLib.h, uma das mais importantes para o projeto, foi desenvolvida pela Open Energy Monitor, possui código aberto, o que significa que está disponível para qualquer usuário que deseje utiliza-la de forma gratuita. Basicamente o algoritmo da biblioteca realiza a contagem de ciclos de um sinal em um determinado período, número de vezes que esse sinal cruza o eixo, de forma a obter o valor mais exato do deslocamento de fase.

As funções da biblioteca *EmonLib.h* são utilizadas por meio de um objeto do tipo *EnergyMonitor*, para a medição de corrente inicia-se o objeto por meio da função “*SCT013.current(pino, calibração)*”, sendo o valor de calibração calculado no item 3.3 para o sensor de corrente. Para o sensor de tensão a função “*emon1.voltage(pino, VOLT_CALL, mudança de fase)*”, no código há a variável de calibração *VOLT_CAL*. Esta variável tem como função fazer uma compensação no cálculo, pois os componentes utilizados na construção de cada sensor podem ter valores ligeiramente diferentes do especificado (considerar a tolerância do componente). Para o protótipo, utilizou-se o valor 211.6, o qual foi obtido por meio de testes com o multímetro, e para o valor de mudança de fase utilizou-se 1.7 (OLIVEIRA, 2018).

Para a leitura de corrente, chama-se a função “*SCT013.calcIrms(1480)*”, em que 1480 é o número de amostras usadas para determinar o valor da corrente. Para a leitura de tensão, chama-se a função “*emon1.calcVI(17, 2000)*”, onde 17 é o número de semiciclos e 2000 o tempo limite para a medição, em seguida a variável tensão recebe o valor de “*emon1.Vrms*”. Para calcular a energia, multiplica-se a corrente medida pela tensão da rede e pelo tempo. Como o loop possui um delay de 1 segundo, a cada iteração multiplica-se a potência encontrada (*Irms* x *tensao*) por 1/3600, de forma a termos o resultado em watt-hora (1 segundo é igual a 1 hora dividido por 3600), como mostrado na Figura 18.

Figura 18 – Método utilizado para calcular o consumo de energia através da potência instantânea.

```
potencia = Irms * tensao;
if (millis() >= tempoanterior + 1000) {
    x = (potencia / 3600); //calcula e guarda em x o valor do consumo;
    consumo = consumo + x; //após 1 segundo, soma e guarda em y o valor de x
    tempoanterior = millis();
}
```

Fonte: imagem elaborada pelo próprio autor (2022).

3.4.1 Lógica e funcionamento

Inicialmente, quando o sistema é ligado é solicitado o ajuste de data e hora do sistema, sendo o ajuste realizado por meio de botões, *up*, *down* e *enter*, como ilustrado na Figura 19.

Figura 19 – Tela de ajuste do horário.



Fonte: Imagem produzida pelo próprio autor (2022).

Após os ajustes no menu principal, o usuário tem acesso a três opções: Monitoramento de dados, onde são exibidas todas as informações de leitura de corrente, tensão e consumo de energia. As imagens das telas são mostradas nas Figura 20 e 21.

Figura 20 – Menu de Monitoramento em estado de entrada.



Fonte: Imagem produzida pelo próprio autor (2022).

Figura 21 - Menu de monitoramento em estado de exibição de grandezas.



Fonte: Imagem produzida pelo próprio autor (2022).

Para programar o ajuste do consumo diário máximo, o usuário deverá utilizar os botões para entrar com o valor diário recomendado para consumo em kWh máximo, conforme mostrado nas Figuras 22 e 23. Por meio de testes realizados na tabela desenvolvida neste

trabalho, verificou-se que a tarifa branca é viável para um consumo de até 12% do total diário dentro do horário de ponta, por esse motivo adotou-se no software esse valor como limite.

Figura 22 – Menu de entrada do valor de consumo máximo diário.



Fonte: Imagem produzida pelo próprio autor (2022).

Figura 23 – Exibição do consumo diário máximo recomendado.



Fonte: Imagem produzida pelo próprio autor (2022).

Caso o usuário atinja esse limite diário o circuito conectado ao relé é desligado e um alerta sonoro é emitido juntamente de uma mensagem no display.

O menu para ajuste de data e hora, é mostrado nas figuras 24 e 25. Nessas telas o usuário tem acesso aos valores de data e hora registradas pelo sistema.

Figura 24 - Menu data e hora.



Fonte: Imagem produzida pelo próprio autor (2022).

Figura 25 – Exibição de data e hora no menu.



Fonte: Imagem produzida pelo próprio autor (2022).

Após ligado e configurado todo o sistema, as variáveis do sistema começam a receber valores. Com o intuito de melhor organizar os dados, o sistema de monitoramento está programado para salvar os valores de consumo apenas a partir das 00:00:01 horas do dia seguinte a sua ativação. Ao abrir o cartão SD do protótipo em um computador o usuário irá encontrar 3 arquivos, consumo.txt, memoria.txt, diario.txt. No arquivo consumo.txt, encontram-se todos os valores de consumo com sua data e hora em intervalos de 15 minutos entre cada uma das leituras. O arquivo nomeado memoria.txt é utilizado para salvar os valores de consumo e caso ocorra queda de energia ou alguma falha no sistema ao reiniciar o a variável consumo receberá o valor salvo dentro deste arquivo. O arquivo diário.txt estão armazenados os valores acumulados de consumo de cada dia.

3.4.2 Tratamento de dados

Para auxiliar o usuário a compreender e modificar seu perfil de consumo diário, foi desenvolvida uma planilha dinâmica com auxílio do *software* Microsoft Excel 365. A Figura 26 mostra imagens desta planilha, onde é possível importar os dados armazenados no SD. Após a importação destes é possível clicar no botão nomeado “calcular” e o sistema realizará o somatório dos valores dos horários de ponta, fora de ponta e intermediário, assim como demonstrar a porcentagem da energia total consumida no horário de pico.

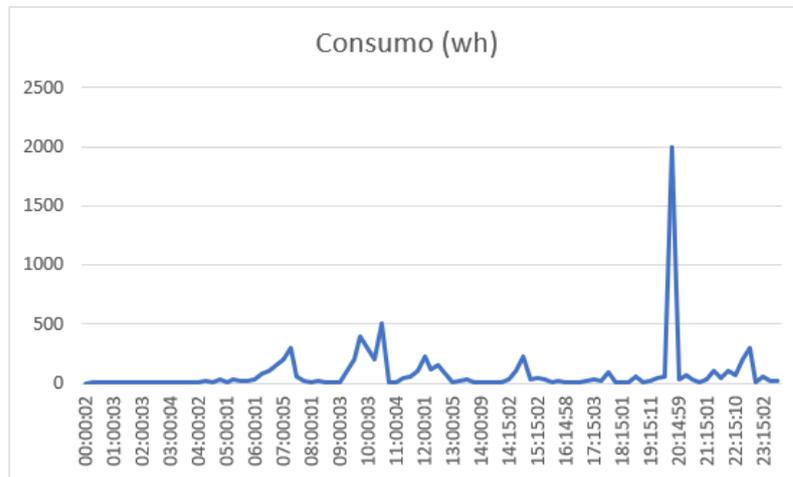
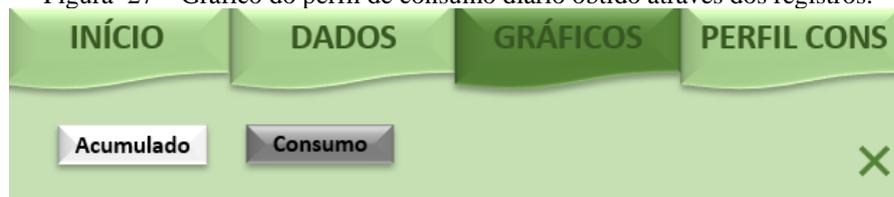
Figura 26 – Página para importação dos dados salvos em arquivo texto no cartão SD.

Data	Hora	Consumo (wh)	Fora de ponta (wh)	Intermediario (wh)	Ponta (wh)
27/05/2022	00:00:02	0	5050	413	2323
27/05/2022	00:15:03	5			
27/05/2022	00:30:01	10			
27/05/2022	00:45:02	5			
27/05/2022	01:00:03	4			
27/05/2022	01:15:04	5			
27/05/2022	01:30:02	10			
27/05/2022	01:45:01	6			
27/05/2022	02:00:03	3			
27/05/2022	02:15:02	5			
27/05/2022	02:30:05	7			
27/05/2022	02:45:02	9			
27/05/2022	03:00:04	3			
27/05/2022	03:15:02	5			
27/05/2022	03:30:01	6			
27/05/2022	03:45:00	10			

Fonte: Imagem produzida pelo próprio autor (2022).

A partir dos dados de consumo, é possível produzir um gráfico com todo o perfil de consumo diário, sendo possível assim uma melhor visualização dos dados, como mostrado na Figura 27. Foi utilizada uma janela de amostragem de 15 minutos, o que diariamente produzirá um total de 96 valores. Assim, foi possível representar estes valores em um gráfico de consumo diário, o qual possibilitará ao usuário verificar quais horários ocorreram maior consumo e se é ainda viável manter-se na “Tarifa Branca”.

Figura 27 – Gráfico do perfil de consumo diário obtido através dos registros.



Fonte: Imagem produzida pelo próprio autor (2022).

Na aba de perfil de consumo o usuário tem acesso a uma tabela com todos os valores das tarifas branca e convencional para os diversos tipos de bandeiras, como mostrado na Figura 28. Os valores utilizados foram retirados de (CEMIG, 2021). Na segunda tabela de dados são realizados os cálculos dos valores diários e mensal para todos os períodos, ponta, fora de ponta e intermediário, baseado nos dados lidos pelo protótipo. Para o resultado da análise, basta o usuário selecionar a bandeira vigente no período a ser analisado, que o sistema retornará o valor estimado a ser pago no atual perfil de consumo para ambas as modalidades tarifárias (branca e convencional), assim como o resultado da análise.

Figura 28 – Página “Perfil de consumo”

	Verde	Amarela	Vermelha 1	Vermelha 2	Escassez hid.
Tarifas					
Ponta	1,17045	1,18919	1,21016	1,26537	1,31245
Intermediário	0,76521	0,78395	0,80492	0,86013	0,90721
Fora de Ponta	0,50898	0,52772	0,54869	0,6039	0,65098
Convencional	0,61805	0,63679	0,65776	0,71297	0,76005
*					
	Fora de ponta (wh)	Interm (wh)	Ponta (wh)	Branca	Convencional
				Escassez hid.	
Diário	5,05	0,413	2,323	R\$ 6,7109	R\$ 5,9177
Mensal	151,5	12,39	69,69	R\$ 201,3284	R\$ 177,5325
*					
Resultado	ALTERAR PERFIL DE CONSUMO				

Gráfico

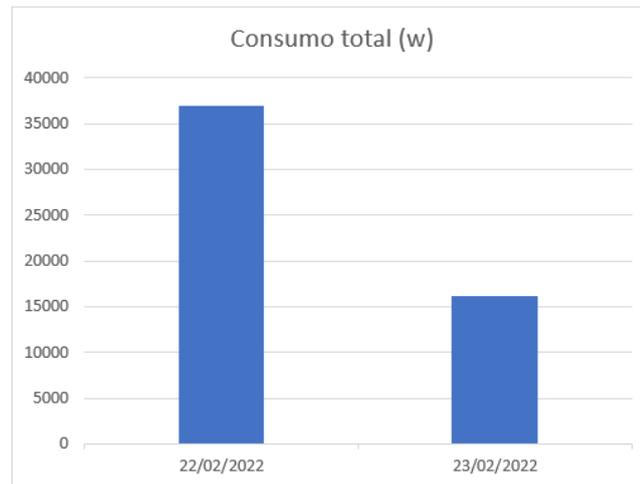
*Valores Kwh CEMIG, para outros estados e concessionárias consultar o site

*Valores apenas para comparações não incluído encargos e taxas

Fonte: Do próprio autor (2022)

No protótipo, também há uma função que grava o somatório diário de consumo da residência, caso o usuário deseje é possível importar esses dados através do menu dados, acumulado. O gráfico dos valores acumulados está disponível no menu gráficos, acumulado, como sugere a Figura 29.

Figura 29 – Análise do perfil de consumo diário registrado pelo sistema embarcado.

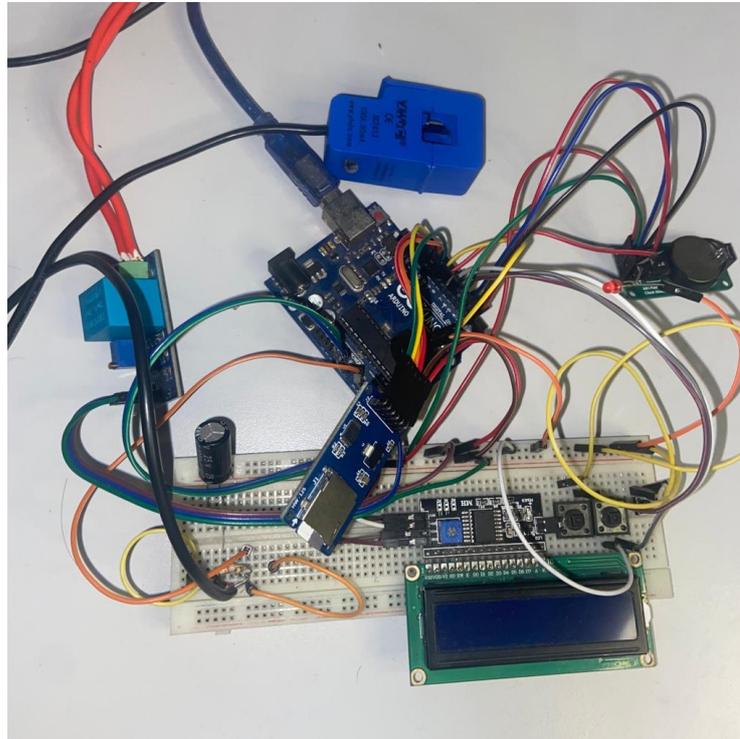


Fonte: Do próprio autor (2022)

4 - RESULTADOS

A montagem física do protótipo proposto, conforme componentes e circuitos definidos na seção anterior, pode ser visualizada pela Figura 30.

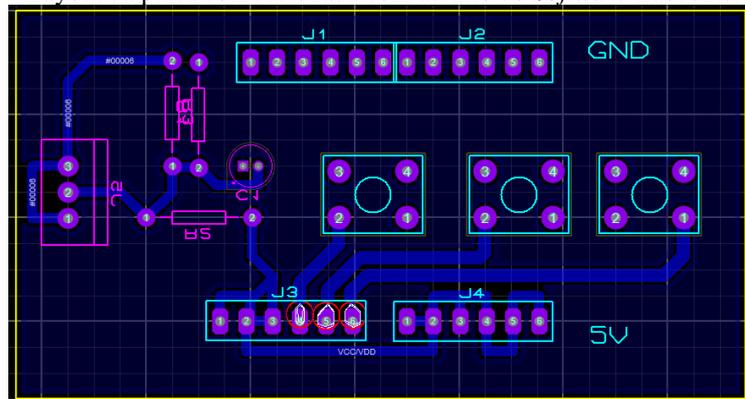
Figura 30 – Protótipo do sistema embarcado para monitoramento e auxílio na adequação de perfil de consumo de energia à tarifa branca



Fonte: Imagem produzida pelo próprio autor (2022).

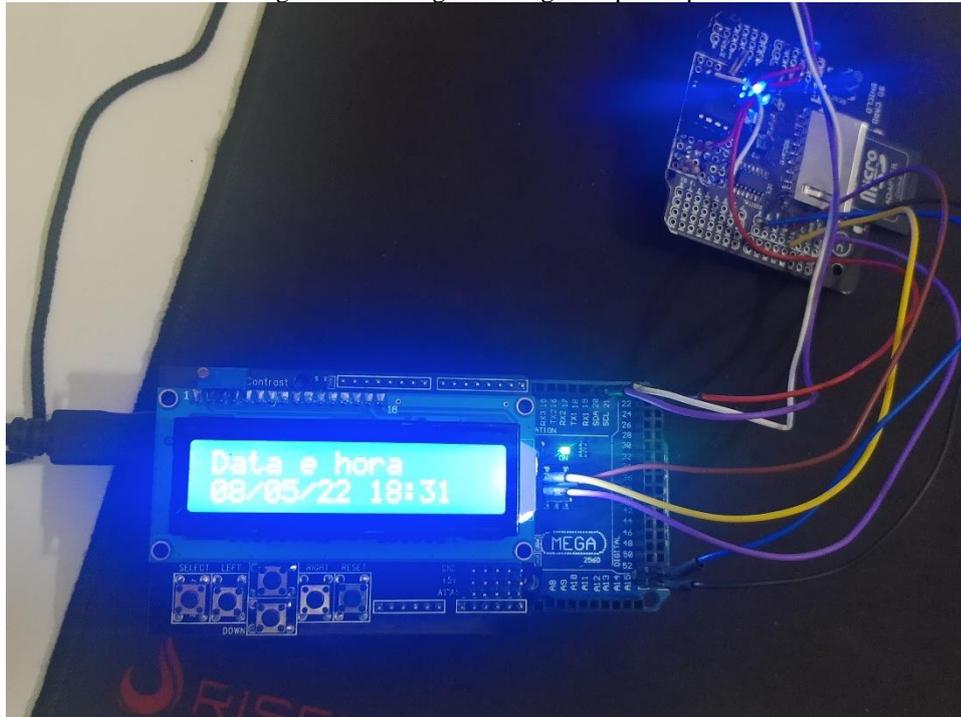
Como a maioria dos dispositivos utilizados são módulos já prontos, foi projetada uma placa para comportar os botões, circuito de carga do SCT-013 e demais dispositivos ao Arduino, como demonstrado na Figura 31 e desenvolvida com auxílio do *software* Labcenter Proteus 8.5. Porém, devido as dificuldades de acesso a laboratórios durante o período de pandemia não foi possível produzir a placa projetada ficando o modelo apresentado nas Figuras 31 e 32 como sugestão para trabalhos futuros.

Figura 31 – Layout da placa confeccionada com auxílio do *Software* Labcenter Proteus 8.5.



Fonte: Imagem produzida pelo próprio autor (2022).

Figura 33 – Imagem do segundo protótipo.



Fonte: Do próprio autor (2022)

Devido ao fato de que o protótipo não pode ter sido testado em laboratório ou em uma residência, alguns testes simples utilizando voltímetro e amperímetro foram realizados, de forma a validar as leituras de tensão e corrente efetuadas pelos sensores. Testes conectando algumas cargas aos sensores, como secadores de cabelo, lâmpadas e dispositivos eletrônicos, de forma a verificar as demais funcionalidades do sistema e tratar possíveis problemas também foram realizados.

Para processar os dados registrados no cartão SD, foi desenvolvida uma planilha dinâmica com auxílio do *software* *Microsoft Excel*. Esta planilha utiliza macros e algumas ações em programação *Visual Basic for Applications (VBA)*, como por exemplo a função de importar os dados txt e inseri-los na planilha. Para os testes da planilha foram utilizados valores de consumo hipotéticos para avaliar cenários diversos do consumo de uma residência com o objetivo de validar adequadamente os resultados do sistema. A base de dados utilizada encontra-se na tabela 2.

Tabela 2 - Base consumo utilizado para análise

Equipamento	Qtd	Potência (w)	Horas ligado	Potência (kWh)
Chuveiro	1	5500	1	5500
Geladeira	1	500	8	4000
Lâmpadas	5	100	5	2500
TV	2	90	4	720
Ar Condicionado	2	1200	4	9600
Total				22320

Fonte: Produzida pelo próprio autor (2022)

Para a primeira avaliação a base de dados foi de um usuário com consumo mais elevado nos horários de 18:00 horas até as 21:00 horas, período considerado pela tarifa branca como horário de pico. Os resultados obtidos estão dispostos na Figura 34.



Fonte: Imagem produzida pelo próprio autor (2022).

Para este perfil de consumo é possível observar que os valores a serem pagos pelo consumidor no modelo da tarifa branca são bem mais altos se comparados com a tarifa convencional, o que geraria um prejuízo ao consumidor de quase R\$ 89,00 por mês na fatura caso a residência estivesse no modelo tarifário branco.

A segunda avaliação dos dados foi realizada com consumo mais elevado nos horários entre 17:00 e 18:00 horas e entre as 21:00 e 22:00 horas, período considerado pela tarifa branca como intermediário. Os resultados obtidos estão dispostos na figura 35.

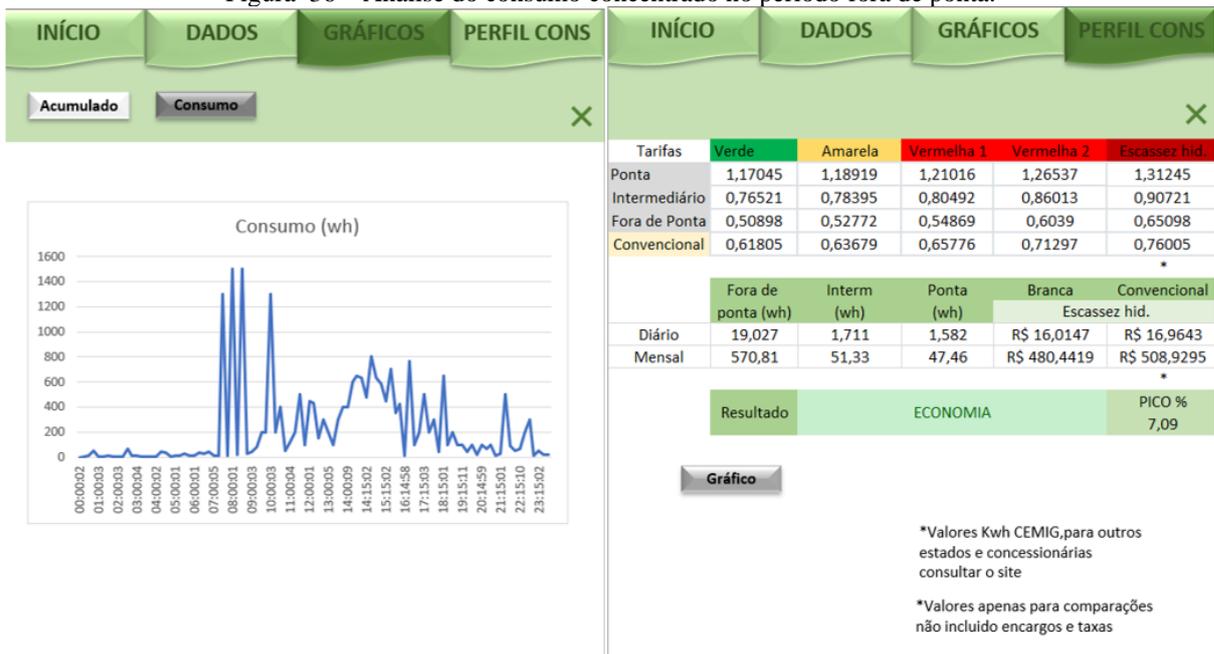


Fonte: Do próprio autor (2022)

Para o mesmo padrão de consumo, porém apenas com uma melhor distribuição dos horários de consumo, retirando os maiores valores do horário de pico e concentrando-os nos horários intermediários se comparadas os resultados da figuras 34 e 35 é possível observar uma economia para o modelo da tarifa branca, porém se esse valor for comparado com o valor a ser pago pela tarifa convencional, ainda temos prejuízo ao utilizar o modelo da tarifa branca.

A terceira avaliação dos dados foi feita com um consumo mais elevado e distribuído nos horários após as 22:00 horas e até as 17:00 horas, período considerado pela tarifa branca como fora de ponta. Os resultados obtidos estão dispostos na Figura 36.

Figura 36 – Análise do consumo concentrado no período fora de ponta.

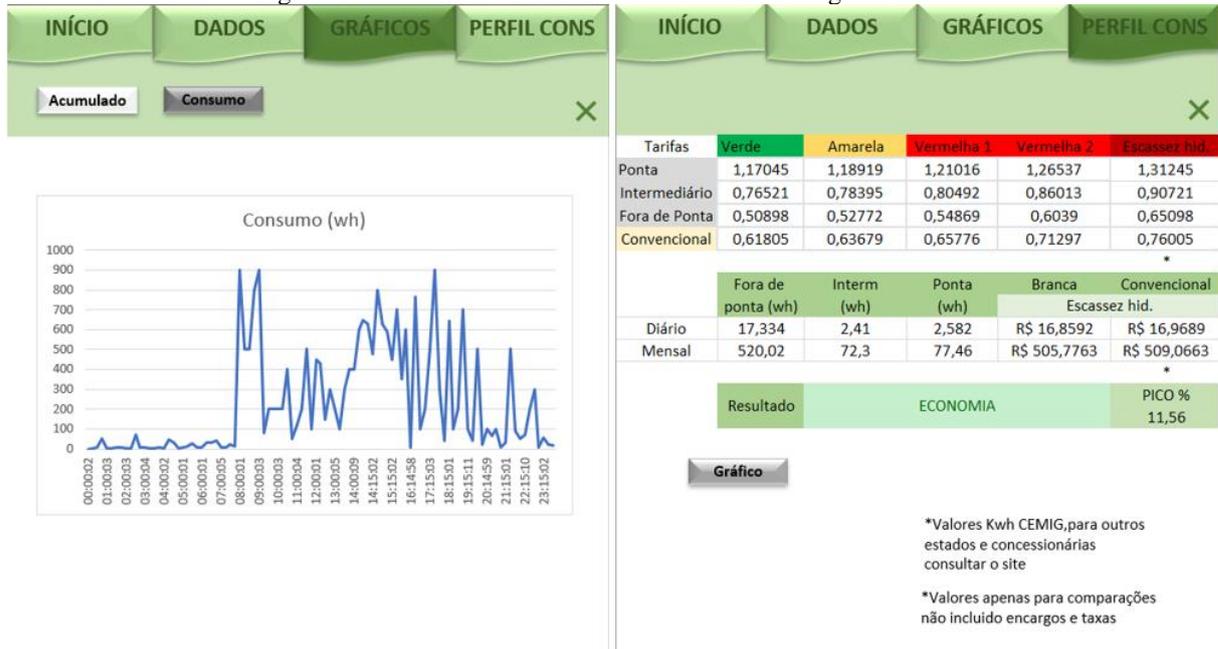


Fonte: Imagem produzida pelo próprio autor (2022).

Ainda seguindo o mesmo padrão, pode-se observar pelos resultados que para os valores de consumo estudados, este seria o perfil de consumo ideal, pois para este padrão, têm-se um desconto de aproximadamente R\$ 28,00 mensal na fatura, se comparado a tarifa convencional.

Também foi realizada uma quarta análise para uma distribuição dos valores de forma mais uniforme durante o período do dia, conforme mostrado na Figura 37. Nesse estudo, foram realizadas diversas variações dos valores seguindo o mesmo padrão descrito, dessa forma, observou-se que para um consumo de até 12% do total diário dentro horário de pico, o modelo da tarifa branca ainda é uma alternativa viável para geral economia na fatura de energia elétrica.

Figura 37 – Análise de resultados distribuídos ao longo de um dia.



Fonte: Do próprio autor (2022).

5 CONCLUSÃO

Ao observar os gráficos da seção de resultados é possível perceber que para os 3 testes realizados apenas alterou-se os valores correspondentes as potências equivalentes a 15 minutos de banho (aproximadamente 1400 kWh), com isso, pode-se concluir que para alguns casos apenas com a troca do horário do banho para o horário fora de ponta já é possível gerar economia com a tarifa branca.

Vários testes alterando-se os dados de consumo na tabela foram realizados, o que se observou com mais frequência durante esses testes foi que a tarifa branca é a alternativa mais viável para a maioria dos casos em que o consumo dentro do horário de pico é de até 12% do total diário, porém, para projetos futuros sugere-se o estudo prático, realizado por meio de testes com o protótipo instalado em residências de modo a verificar se esse padrão de 12% se mantém para valores reais de consumo.

A montagem do circuito foi bem sucedida na fase de prototipação, porém o sistema ainda encontra-se em *proto board*, ficando como uma segunda sugestão de trabalho futuro a montagem do protótipo em placa de circuito impresso para melhor instalação e testes como sugere a figura 16 da seção 3.2. Outras melhorias ainda podem ser implementadas, como por exemplo a lógica de controle de desligamento de circuitos que, devido a impossibilidade da realização de testes não foi muito desenvolvida nesta versão do protótipo.

Portando, a análise dos dados através das leituras efetuadas pelo protótipo é de extrema importância pois somente com ela será possível um estudo personalizado para cada caso e com isso tirar melhor proveito da tecnologia e gerar maiores economias na fatura de eletricidade.

REFERÊNCIAS

- ABRADEE. **Tarifa Branca.** Disponível em: <https://www.neoenergiaelektro.com.br/Media/Default/pdf/TARIFA-BRANCA_abradee.pdf>. Acesso em: 30 maio. 2022.
- ALMEIDA, J. Á. J. DE. **P&D no setor elétrico brasileiro: um estudo de caso na companhia hidro elétrica do São Francisco.** [s.l.] Universidade Federal de Pernambuco, 2008.
- ANEEL Agência Nacional de Energia Elétrica. Com acréscimo de 601,5 MW, maio tem a maior expansão da geração em 2022. Disponível em: <encurtador.com.br/aeQU9> Acesso em: 11 de jul. 2022.
- ANDRADE, R. B. DE. **A nova crise hídrica e as lições do racionamento em 2001.** Disponível em: <<https://fieac.org.br/index.php/imprensa/artigos/534-a-nova-crise-hidrica-e-as-licoes-do-acionamento-de-2001.html>>. Acesso em: 31 maio. 2022.
- BITTENCOURT, I. R.; MARTINS, G. C.; GASTALDELLO, D. S. Desenvolvimento de Medidor de Energia Cognitivo para Monitoramento de Consumo Energético Residencial. **Revista e-F@tec**, v. 11, n. 1, 2021.
- CAMARGO, L. T. **Projeto de Sistemas Fotovoltaicos conectados à Rede Elétrica.** Londrina: Trabalho de Conclusão de Curso Universidade Estadual de Londrina, 2017.
- CEMIG. **Bandeira Tarifária.** Disponível em: <<https://www.cemig.com.br/atendimento/bandeira-tarifaria/>>. Acesso em: 30 maio. 2022.
- CORSINI, I. **Energia elétrica aumento mais do que o dobro da inflação nos últimos anos.** Disponível em: <<https://www.cnnbrasil.com.br/business/energia-eletrica-aumentou-mais-do-que-o-dobro-da-inflacao-nos-ultimos-anos/>>. Acesso em: 30 maio. 2022.
- CPFL. **Bandeiras Tarifárias.** Disponível em: <<https://www.cpfl.com.br/bandeiras-tarifarias>>. Acesso em: 13 jun. 2022.
- DE BRITO, J. L. G. **Sistema para monitoramento de consumo de energia elétrica particular , em tempo real e não invasivo utilizando a tecnologia Arduino.** Londrina - PR: Trabalho de Conclusão de Curso Universidade Estadual de Londrina, 2016.
- ELÉTRICA, A. N. DE E. **Geração Distribuída.** Disponível em: <<https://www.gov.br/aneel/pt-br/assuntos/geracao-distribuida#:~:text=Desde 17 de abril de,de distribuiçao de sua localidade.>>. Acesso em: 13 jun. 2022a.
- ELÉTRICA, A. N. DE E. **Tarifa Branca.** Disponível em: <<https://www.gov.br/aneel/pt-br/assuntos/tarifas/tarifa-branca#:~:text=A Tarifa Branca é uma,Rural%3A denominada subgrupo B2%3B e>>. Acesso em: 27 maio. 2022b.
- ENERGÉTICA, E. DE P. **Balanco Energético Nacional.** Rio de Janeiro: Empresa de Pesquisa Energética, 2020a.
- ENERGÉTICA, E. DE P. **Balanco Energético Nacional.** Rio de Janeiro: Empresa de Pesquisa Energética, 2020b.
- ENERGÉTICA, E. DE P. **Anuário Estatístico de Energia Elétrica 2021.** Rio de Janeiro: [s.n.].
- ENERGIA, W. **Entenda os Encargos Setoriais.** Disponível em: <<https://www.witzler.com.br/2021/09/09/entenda-os-encargos-setoriais/>>. Acesso em: 1 jun. 2022.
- IDEC. **Encargos e tributos compõe mais de 40% da sua conta de luz.** Disponível em: <https://idec.org.br/edasaconta_bkp/encargosetributos>. Acesso em: 1 jun. 2022.
- INFO, U. **Sensor de Tensão ZMPT101B.**
- LEARN OPEN ENERGY MONITOR. **CT Sensors - Interfacing wigh an Arduino.** Disponível em: <<https://learn.openenergymonitor.org/electricity-monitoring/ct>>

sensors/interface-with-arduino>. Acesso em: 13 jun. 2022.

MALAR, J. P. **Tarifa Branca pode reduzir conta de luz em até 15% mas exige cuidado**. Disponível em: <[https://www.cnnbrasil.com.br/business/tarifa-branca-pode-reduzir-conta-de-luz-em-ate-15-mas-exige-cuidado/#:~:text=Se ele aderir à Tarifa,ele conseguirá reduzir sua conta.](https://www.cnnbrasil.com.br/business/tarifa-branca-pode-reduzir-conta-de-luz-em-ate-15-mas-exige-cuidado/#:~:text=Se%20ele%20aderir%20%C3%A0%20Tarifa,ele%20conseguir%C3%A1%20reduzir%20sua%20conta.)>. Acesso em: 1 jun. 2022.

OLIVEIRA, B. A. S. Desenvolvimento de um protótipo de sistema de monitoramento de energia elétrica via internet. v. 12, n. 1, p. 48–61, 2018.

ONS. Mapa dinâmico do SIN. Disponível em: <<http://www.ons.org.br/paginas/sobre-o-sin/mapas>>. Acesso em: 11 de jul. 2022.

PAHO. **Histórico da pandemia de COVID-19**. Disponível em: <<https://www.paho.org/pt/covid19/historico-da-pandemia-covid-19>>. Acesso em: 14 jun. 2022.

PEREIRA, D. A. T. et al. **Automação Residencial: comando remoto de cargas residenciais através de plataforma própria com comunicação Wireless**. XVIII Conferência de Estudos em Engenharia Elétrica. **Anais...2020**

RENOVÁVEIS, E. ENERGIAS. **Energia Solar fotovoltaica 2021: expectativa e projeções**. Disponível em: <<https://www.ecoaenergias.com.br/energia-fotovoltaica-crescimento-2021/>>. Acesso em: 31 maio. 2022.

ROSELLI, M. A. et al. **Análise diagnóstica dos efeitos da tarifa branca utilizando data analytics**. XII Congresso Brasileiro de Regulação e 6ª Expo Abar. **Anais...Foz do Iguaçu - PR: 2021** Disponível em: <https://www.researchgate.net/profile/Marcio-Andrey-Roselli/publication/360691905_ANALISE_DIAGNOSTICA_DOS_EFEITOS_DA_TARIFA_BRANCA_UTILIZANDO_DATA_ANALYTICS/links/62858fbd50c4566fc2744cb5/ANALISE-DIAGNOSTICA-DOS-EFEITOS-DA-TARIFA-BRANCA-UTILIZANDO-DATA-ANA>

SICARI, L. F. **Ambientes de contratação de energia Mercado Livre x Mercado Cativo**. [s.l.] Universidade Federal de Uberlândia, 2020.

TONELLI, I. M. **Desenvolvimento de um medidor de energia integrado à Internet das coisas para gestão do lado da demanda**. Araranguá: Universidade Federal de Santa Catarina, 2019.

APÊNDICE A

Código implementado no Protótipo 1.

```
//Trabalho de conclusão de curso aluno Celso Oliveira
//Data: 08/05/2022
//=====Bibliotecas
Utilizadas=====
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#include <DS1302.h>
#include <SPI.h>
#include <SD.h>
#include "EmonLib.h"
//=====
=====
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

DS1302 rtc(2, 3, 4); // Pinos (RST, DATA, CLK)

//=====Biblioteca
Emonlib=====
EnergyMonitor SCT013;
int pinSCT = A0; //Pino analógico conectado ao SCT-013
#define VOLT_CAL 211.6 //VALOR DE CALIBRAÇÃO
EnergyMonitor emon1; //CRIA UMA INSTÂNCIA

File arquivo;

//=====Variaveis
Utilizadas=====
float potencia = 0;
int porta_rele1 = 8;
int buzzer = 9;
float consmax = 0;
float consumo = 0;
float consumod = 0;
float consumoh = 0;
float diario = 0;
float memoria = 0;
float conspico = 0;
float consdia = 0;
float diarioi = 0;
float diariof = 0;
float diariop = 0;
int picol = 0;
float x = 0;
float y = 0;
long tempoanterior = 0;
long tempod = 0;
long tempomem = 0;
long tempovr = 0;
int hora = 0;
int minuto = 0;
int segundo = 0;
int dia = 0;
int mes = 0;
int ano = 2022;
int timeadj = 0;
```

```

int dataadj = 0;
int modo = 0;

//=====Variaveis
Menu=====

#define botao_MAIS 5 //define o botão no pino 2 (MAIS)
#define botao_ENTER 6 //define o botão no pino 3
(ENTER/VOLTAR)
#define botao_MENOS 7 //define o botão no pino 4 (MENOS)
bool L_botao_MAIS = 1; //variável para fazer a leitura do
botão (MAIS)
bool LA_botao_MAIS = 1; //variável para verificar a leitura
anterior do botão (MAIS)
bool L_botao_ENTER = 1; //variável para fazer a leitura do
botão (ENTER/VOLTAR)
bool LA_botao_ENTER = 1; //variável para verificar a leitura
anterior do botão (ENTER/VOLTAR)
bool L_botao_MENOS = 1; //variável para fazer a leitura do
botão (MENOS)
bool LA_botao_MENOS = 1; //variável para verificar a leitura
anterior do botão (MENOS)
int tela = 1; //variável para controlar a mudança de tela no menu
int addr = 0x00;

void setup() {
  pinMode(buzzer, OUTPUT);
  // Ajusta o clock para o modo de execução e desativa contra proteção
  Serial.begin(9600);
  rtc.halt(false);
  rtc.writeProtect(false);

  //set das datas iniciais-----
  //rtc.setDOW(SABADO); // Dia da Semana
  rtc.setTime(17, 58, 50); // Horário 12:00:00 (formato 24hr)
  //rtc.setDate(6, 5, 2022); // Data (dia, mês, ano)

  pinMode(botao_MAIS, INPUT_PULLUP); //define o botao (MAIS) como uma
  entrada pull up
  pinMode(botao_ENTER, INPUT_PULLUP); //define o botao (ENTER/VOLTAR) como
  uma entrada pull up
  pinMode(botao_MENOS, INPUT_PULLUP); //define o botao (MENOS) como uma
  entrada pull up
  lcd.begin(16, 2); //inicia o display 16x2
  lcd.clear(); //limpa todos os caracteres do lcd
  lcd.setCursor(0, 0); //posiciona o cursor no display
(caracter, linha)
  lcd.print(F(" SEJA BEM VINDO ")); //imprime o texto entre aspas
  lcd.setCursor(0, 1); //posiciona o cursor no display
(caracter, linha)
  lcd.print(F("Ini SD card... ")); //imprime o texto entre aspas
  delay(3000);

  lcd.setCursor(0, 1);
  if (!SD.begin(10)) {
    lcd.print(F(" Falha no SD "));
    while (1);
  } lcd.print(F(" Sucesso! "));
  delay(3000); //pausa o programa por 3 segundos

```

```

arquivo = SD.open("memoria.txt"); // Abre o Arquivo memoria
consumo = arquivo.parseFloat();
arquivo.close();

SCT013.current(pinSCT, 60.606); //60.606 número total de voltas (espiras)
do secundário pelo valor do resistor de carga dimensionado:
emon1.voltage(2, VOLT_CAL, 1.7); //PASSA PARA A FUNÇÃO OS PARÂMETROS
(PINO ANALÓGICO / VALOR DE CALIBRAÇÃO / MUDANÇA DE FASE)
lcd.clear();
}

void loop() {
  // === Ajuste data === //-----
  -----
  if (timeadj == 0 && dataadj == 0) {
    lcd.setCursor(0, 0);
    lcd.print(F(" Ajuste Hora:      "));
    lcd.setCursor(5, 1);
    lcd.print(hora);
    lcd.print(":");
    lcd.print(minuto);
    lcd.print(":");
    lcd.print(segundo);
    if (digitalRead(botao MAIS) == 0) {
      hora += 1;
      if (hora > 23) hora = 0;
      rtc.setTime(hora, minuto, 0);
      while (digitalRead(botao MAIS) == 0) {}
    }
    if (!digitalRead(botao MENOS)) {
      minuto += 1;;
      if (minuto > 59) minuto = 0;
      rtc.setTime(hora, minuto, 0);
      while (!digitalRead(botao MENOS)) {}
    }
    if (digitalRead(botao ENTER) == 0) {
      dataadj = 1;
      delay(1000);
    }
  }
  if (dataadj == 1 && timeadj == 0) {
    lcd.setCursor(0, 0);
    lcd.print(F(" Ajuste Data:      "));
    lcd.setCursor(4, 1);
    lcd.print(dia);
    lcd.print("/");
    lcd.print(mes);
    lcd.print("/");
    lcd.print(ano);
    if (digitalRead(botao MAIS) == 0) {
      dia += 1;
      if (dia > 31) dia = 0;
      rtc.setDate(dia, mes, ano);
      while (digitalRead(botao MAIS) == 0) {}
    }
    if (!digitalRead(botao MENOS)) {
      mes += 1;;
      if (mes > 12) mes = 0;
      rtc.setDate(dia, mes, ano);
    }
  }
}

```

```

    while (!digitalRead(botao_MENOS)) {}
  }
  if (digitalRead(botao_ENTER) == 0) {
    timeadj = 1;
  }
}
if (timeadj == 1 && dataadj == 1) {

  // === Verifica se Sd está funcionando corretamente durante o periodo
de funcionamento === //-----
  -----
  if (!SD.begin(10)) {
    lcd.setCursor(0, 0);
    lcd.print(F(" Falha no SD      "));
    lcd.setCursor(0, 1);
    lcd.print(F(" Reiniciar      "));
    while (1);
  } else {

    double Irms = SCT013.calcIrms(1480); //atribui o valor de corrente
lido a variavel irms

    emon1.calcVI(17, 2000); //FUNÇÃO DE CÁLCULO (17 SEMICICLOS, TEMPO
LIMITE PARA FAZER A MEDIÇÃO)
    float tensao = emon1.Vrms; //VARIÁVEL RECEBE O VALOR DE TENSÃO RMS
OBTIDO
    if (tensao < 10) { // correção de erro do sensor
      tensao = 0;
    }
    potencia = Irms * tensao; // Calcula o valor da Potencia
Instantanea
    if (Irms < 0.04) {
      Irms = 0; potencia = 0;
    }
    if (millis() >= tempoanterior + 1000) {
      x = (potencia / 3600); //calcula e guarda em x o valor do consumo;
de x
      consumo = consumo + x; //após 1 segundo, soma e guarda em y o valor
      consumod = consumod + x;
      consumoh = consumoh + x;
      tempoanterior = millis();
    }

    // === Desligamento forçado //-----
    -----
    String t = rtc.getTimeStr();
    consmax = consmax * 1000;
    conspico = consdia * 0.12;
    Serial.println(t);
    if (t == "18:00:00") {
      picol = 1;
    }
    if (picol == 1) {
      diariof = 0;
      diarioi = 0;
      if (millis() >= tempovr + 1000) {
consumo;
        x = (potencia / 3600); //calcula e guarda em x o valor do
        diariof = diariof + x;

```

```

    tempovr = millis();
    Serial.println(diariop);
    Serial.println(consdia);
}
if (diariop > conspico) {
    tone(buzzer, 5000);
    lcd.setCursor(0, 1);
    lcd.print(F("Consumo limite          ")); // Imprime Horário
    //Desligando o buzzer.
    noTone(buzzer);
    delay(500);
    digitalWrite(porta_rele1, HIGH); //Liga rele 1
} else {
    digitalWrite(porta_rele1, LOW); //Liga rele 1
}
}

// Salvar dados cartão SD-----
-----
if (t == "00:00:01") {
    arquivo = SD.open("diario.txt", FILE_WRITE);
    if (arquivo) {
        arquivo.print(rtc.getDateStr());
        arquivo.print(';');
        arquivo.print(rtc.getTimeStr());
        arquivo.print(';');
        arquivo.println(consumod); //acumulado geral
        arquivo.close();
        consumod = 0;
    } else {
        lcd.setCursor(0, 0);
        lcd.print("  error opening          ");
        lcd.setCursor(0, 1);
        lcd.print(F("  test.txt          "));
    }
}
if (t == "00:00:01") {
    modo = 1;
}
if (modo == 1) {
    if (millis() >= temposd + 60000) { // utilizar 3600000 para horas
        arquivo = SD.open("tarifas.txt", FILE_WRITE);
        if (arquivo) {
            arquivo.print(rtc.getDateStr());
            arquivo.print(';');
            arquivo.print(rtc.getTimeStr());
            arquivo.print(';');
            arquivo.println(consumoh); //acumulado geral
            arquivo.close();
            consumoh = 0;
        } else {
            lcd.setCursor(0, 0);
            lcd.print("  error opening          ");
            lcd.setCursor(0, 1);
            lcd.print(F("  test.txt          "));
        }
        temposd = millis();
    }
}
}

```

```

// === Memória do sistema === //-----
-----
if (millis() >= tempomem + 1000) { // MEMORIA DE CONSUMO
  SD.remove("memoria.txt");
  arquivo = SD.open("memoria.txt", FILE_WRITE);
  if (arquivo) {
    arquivo.println(consumo); //acumulado geral
    arquivo.close();
  } else {
    lcd.setCursor(0, 0);
    lcd.print("  error opening  ");
    lcd.setCursor(0, 1);
    lcd.print(F("  memoria.txt  "));
  }
  tempomem = millis();
}
// === Ações no display LCD === //-----
-----
if (tela == 1) { //se a variável tela for igual
a MAIS, faça...
  lcd.setCursor(0, 0); //posiciona o cursor no
display (caracter, linha)
  lcd.print(F("  << MENU >>  ")); //imprime o texto entre
aspas
  lcd.setCursor(0, 1); //posiciona o cursor no
display (caracter, linha)
  lcd.print(F(">Monitoramento  ")); //imprime o texto entre
aspas
}
else if (tela == 2) {
  lcd.setCursor(0, 0);
  lcd.print(F("  << MENU >>  "));
  lcd.setCursor(0, 1);
  lcd.print(F(">Adj Cons Max  "));
}
else if (tela == 3) {
  lcd.setCursor(0, 0);
  lcd.print(F("  << MENU >>  "));
  lcd.setCursor(0, 1);
  lcd.print(F(">Data e Hora  "));
}
}
else if (tela == 10) {
  lcd.setCursor(0, 0);
  lcd.print("I:");
  lcd.print(Irms);
  lcd.print("A ");
  lcd.setCursor(8, 0);
  lcd.print("U:");
  lcd.print(tensao);
  lcd.print("V");
  lcd.setCursor(0, 1);
  lcd.print("Consumo: ");
  lcd.print(consumo, 2);
  lcd.print(" W  ");
}
}
else if (tela == 20) {
  lcd.setCursor(0, 0);
  lcd.print(F("  Adj consumo  "));
  lcd.setCursor(0, 1);

```

```

lcd.print(F("    "));
lcd.setCursor(4, 1);
lcd.print(consmax);
lcd.print(F(" kW/h "));
if (digitalRead(botao_MAIS) == 0) {
    consmax += 1;
    while (digitalRead(botao_MAIS) == 0) {}
}
if (!digitalRead(botao_MENOS)) {
    consmax -= 1;;
    while (!digitalRead(botao_MENOS)) {}
}
}
else if (tela == 30) {
    lcd.setCursor(0, 0);
    lcd.print(rtc.getDateStr()); // Imprime a data
    lcd.setCursor(10, 0);
    lcd.print(("    "));
    lcd.setCursor(0, 1);
    lcd.print(rtc.getTimeStr()); // Imprime Horário
    lcd.setCursor(8, 1);
    lcd.print(("    "));
}

////////////////////////////////////

// === Ações do Botão MAIS === // //////////////////////////////////
L_botao_MAIS = digitalRead(botao_MAIS);
if (!L_botao_MAIS && LA_botao_MAIS) {
    if (tela == 1)tela = 2;
    else if (tela == 2)tela = 3;
    else if (tela == 3)tela = 1;
    delay(30);
}
LA_botao_MAIS = L_botao_MAIS;

////////////////////////////////////

// === Ações do Botão MENOS === // //////////////////////////////////
L_botao_MENOS = digitalRead(botao_MENOS);
if (!L_botao_MENOS && LA_botao_MENOS) {
    if (tela == 1)tela = 3;
    else if (tela == 2)tela = 1;
    else if (tela == 3)tela = 2;
    delay(30);
}
LA_botao_MENOS = L_botao_MENOS;

////////////////////////////////////

// === Ações do Botão ENTER === // //////////////////////////////////
L_botao_ENTER = digitalRead(botao_ENTER);
if (!L_botao_ENTER && LA_botao_ENTER) {
    if (tela == 1)tela = 10;
    else if (tela == 2)tela = 20;
    else if (tela == 3)tela = 30;
    else if (tela == 10)tela = 1;
    else if (tela == 20)tela = 2;
}

```

```

        else if (tela == 30)tela = 3;
        delay(30);
    }
    LA_botao_ENTER = L_botao_ENTER;

}
}
}

```

APÊNDICE B

Código implementado no Protótipo 2.

```

/*
 * Trabalho de conclusão de curso aluno Celso Oliveira
 * Data: 08/05/2022
 *
 *=====Pinagem display LCD=====
 * PINO MÓDULO LCD ----- PINO ARDUINO:
 * RS ----- D8
 * ENABLE ----- D9
 * LCD DATA 4 ----- D4
 * LCD DATA 5 ----- D5
 * LCD DATA 6 ----- D6
 * LCD DATA 7 ----- D7
 *
 * =====Pinagem do Circuito para o cartão SD no Arduino Mega=====
 * PINO MÓDULO SD ----- PINO ARDUINO:
 * MOSI ----- D51 - ICSP4
 * MISO ----- D50 - ICSP1
 * CLK ----- D52 - ICSP 3
 * CS ----- D53
 */

/*
 *===== Circuito para o DS1307 com o protocolo Wire:=====
 * PINO MÓDULO RTC DS1307 ----- PINO ARDUINO:
 * SDA ----- D20
 * SC ----- D21
 */

//Inclusão de bibliotecas

#include <EmonLib.h>
#include <TimeLib.h> // Biblioteca tempo
#include <Wire.h> //Biblioteca I2C
#include <DS1307RTC.h> //Inclui o RTC
#include <LiquidCrystal.h> //Biblioteca Display LCD
#include <SPI.h>
#include <SD.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7); //Cria objeto para configurar o display LCD e usa os
pinos digitais em parenteses para acionar o LCD
DS1307RTC rtc;
tmElements_t data; //Estrutura para ajustar o tempo de escrita ou leitura do RTC
SDClass sd; //Cria objeto para o cartão de memória SD
EnergyMonitor emon1;

//Define a constante para configurar o pino CS para leitura do cartão SD
const int cartao_cs = 53;

volatile int leitura = 0; //Variável global para leitura da entrada analógica A0 que está
ligada aos botões do menu

```

```

/*=====Variáveis globais para ajuste de data e
hora=====*/
volatile int dia = 0;
volatile int mes = 0;
volatile int ano = 0;
volatile int hora = 0;
volatile int minuto = 0;
volatile int segundo = 0;

/*===== Variáveis globais para ajustar a tela do LCD
=====*/
volatile int flag_tela = 0; //Variaveis locais para mudança de tela no menu
volatile int tela_ajustar_rtc = 0; //Variavel global para permitir a entrada na tela de ajuste
do horario do RTC
volatile int tela_modos_operacao = 0; //Variavel global para escolher entre o modo Logger ou
modo de tomada de decisão
volatile int tempo_tela = 30000; //Define o tempo de 30s no qual cada tela permanece se nao
tiver decisão

volatile float energia_antes = 0.0000000;
volatile float energia = 0.0000000;
volatile float potencia_real = 0.00;
volatile int janela_amostragem = 14; //Janela de amostragem em minutos
volatile int diferenca_horario = 0;
volatile int horario_anterior = 0;
volatile int horario_atual = 0;

void setup()
{ //=====Início
void
setup()=====//
  // Inicia o display LCD;
  emon1.voltage(4,234.56, 1.7); //Configura o objeto para monitorar tensão pela entrada
analógica A4
  emon1.current(3,111.1); //Configura o objeto para monitorar tensão pela entrada analógica A3
  rtc.read(data);
  dia = data.Day;
  mes = data.Month;
  ano = tmYearToCalendar(data.Year);
  if(ano < 2022) ano = 2022;
  data.Year = CalendarYrToTm(ano);
  hora = data.Hour;
  minuto = data.Minute;
  segundo = data.Second;
  rtc.write(data);

  delay(500);
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.setCursor(0, 0); //posiciona o cursor no display (caracter, linha)
  lcd.print(" SEJA BEM VINDO "); //imprime o texto entre aspas
  lcd.setCursor(0, 1); //posiciona o cursor no display (caracter, linha)
  delay(2000);

  //Tela de ajuste do RTC
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Ajustar RTC?");
  lcd.setCursor(0,1);
  lcd.print("Press Up ou Sel.");
  delay(2000);
  lcd.setCursor(0,1);
  lcd.print(" ");

  int tempo_atual = 0;
  int tempo_anterior = millis();
  while(flag_tela <1)
  { //Início while de seleção para entrar na tela de ajuste

      lcd.setCursor(0,1);
      lcd.print(" ");
      lcd.setCursor(0,1);
      if(tela_ajustar_rtc == 1){
        lcd.print("NAO");
      }else if(tela_ajustar_rtc == 0){

```

```

        lcd.print("SIM");
    }
    leitura = analogRead(A0);
    if((leitura > 140)&&(leitura <=150))//Verifica a leitura do botao Up
    {
        //=== Início do Debounce da chave up === //
        while((leitura > 140)&&(leitura <=150)){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        tela_ajustar_rtc++;
        if(tela_ajustar_rtc >1){tela_ajustar_rtc = 0;}
        flag_tela = 0;
    }else if((leitura >700)&&(leitura <= 750)) //Verifica a leitura do botão Select
    {
        //=== Início do Debounce da chave up === //
        while((leitura >700)&&(leitura <= 750)){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        lcd.clear();
        lcd.setCursor(0,0);
        flag_tela = 1;
    }
    delay(200);

    //Lógica do Timer para ajuste de tela
    tempo_atual = millis();
    if((tempo_atual - tempo_anterior)>=tempo_tela){
        tela_ajustar_rtc = 1;
        flag_tela = 3;
    }

} //Fim while de seleção para entrar na tela de ajuste

lcd.clear();

//===== Tela para configuração do horário do RTC
//=====
tempo_anterior = millis();
if(tela_ajustar_rtc == 0)
{ //Início If ajustar RTC inicial

    //Tela de ajuste do RTC
    flag_tela = 0; //Zera o contador da tela do LCD
    lcd.setCursor(0,0);
    lcd.print("Ajuste da data");

    while(flag_tela <3)
    { //Início while de ajuste da tela
        switch(flag_tela)
        {
            case 0:
                leitura = analogRead(A0);
                if((leitura > 140)&&(leitura <=150)){
                    //=== Início do Debounce da chave up === //
                    while((leitura > 140)&&(leitura <=150)){leitura = analogRead(A0);}
                    delay(5);// Debounce da chave
                    //=== Fim do Debounce da chave up === //
                    dia++;
                    if(dia >31) dia=31;
                }else if((leitura >300)&&(leitura <= 340)){
                    //=== Início do Debounce da chave up === //
                    while((leitura >300)&&(leitura <= 340)){leitura = analogRead(A0);}
                    delay(5);// Debounce da chave
                    //=== Fim do Debounce da chave up === //
                    dia--;
                    if(dia <=1) dia=1;
                }else if (leitura <= 5){
                    //=== Início do Debounce da chave up === //
                    while(leitura <=5){leitura = analogRead(A0);}
                    delay(5);// Debounce da chave
                    //=== Fim do Debounce da chave up === //
                    flag_tela++;
                }
                //imprime a tela piscando o campo dia

```

```

        lcd.setCursor(0,1);
        lcd.print("      ");
        lcd.setCursor(0,1);
        lcd.print(" ");
        lcd.print("/");
        if(mes <10)lcd.print("0");
        lcd.print(mes);
        lcd.print("/");
        lcd.print(ano);
break;
case 1:
    leitura = analogRead(A0);
    if((leitura > 140)&&(leitura <=150)){
        //=== Início do Debounce da chave up === //
        while((leitura > 140)&&(leitura <=150)){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        mes++;
        if(mes >12) mes=1;
        }else if((leitura >300)&&(leitura <= 340)){
        //=== Início do Debounce da chave up === //
        while((leitura >300)&&(leitura <= 340)){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        mes--;
        if(mes <=1) mes=1;
        }else if (leitura <= 20){
        //=== Início do Debounce da chave up === //
        while(leitura <= 20){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        flag_tela++;
    }
    //imprime a tela piscando o campo mes
    lcd.setCursor(0,1);
    lcd.print("      ");
    lcd.setCursor(0,1);
    if(dia <10)lcd.print("0");
    lcd.print(dia);
    lcd.print("/");
    lcd.print(" ");
    lcd.print("/");
    lcd.print(ano);
break;
case 2:
    leitura = analogRead(A0);
    if((leitura > 140)&&(leitura <=150)){
        //=== Início do Debounce da chave up === //
        while((leitura > 140)&&(leitura <=150)){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        ano++;
        if(ano >= 2042){
            ano = 2042;
        }
        }else if((leitura >300)&&(leitura <= 340)){
        //=== Início do Debounce da chave up === //
        while((leitura >300)&&(leitura <= 340)){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        ano--;
        if(ano <= 2022) ano = 2022;
        }else if (leitura <= 20){
        //=== Início do Debounce da chave up === //
        while(leitura <= 20){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        flag_tela = 0;
    }
    //imprime a tela piscando o campo ano
    lcd.setCursor(0,1);
    lcd.print("      ");
    lcd.setCursor(0,1);
    if(dia <10)lcd.print("0");
    lcd.print(dia);
    lcd.print("/");

```

```

        if(mes <10)lcd.print("0");
        lcd.print(mes);
        lcd.print("/");
        lcd.print(" ");
    break;
}

    delay(50);
    lcd.setCursor(0,1);
    lcd.print(" ");
    lcd.setCursor(0,1);
    if(dia <10)lcd.print("0");
    lcd.print(dia);
    lcd.print("/");
    if(mes <10)lcd.print("0");
    lcd.print(mes);
    lcd.print("/");
    lcd.print(ano);
    delay(50);

//Condição para sair da tela de ajuste pressionando o botao Select
if((leitura >700)&&(leitura <= 750)){
    while((leitura >700)&&(leitura <= 750)){leitura = analogRead(A0);}
    delay(5);// Debounce da chave
    flag_tela = 3;
}

//Lógica do Timer para ajuste de tela
tempo_atual = millis();
if((tempo_atual - tempo_anterior)>=tempo_tela){
    flag_tela = 3;
}

} //Fim while de ajuste da tela

    flag_tela = 0;//reinicia flag de estado da tela de ajuste do horario

tempo_anterior = millis(); //Timer para esperar a mudanca de tela;
while(flag_tela <3)
{ //Inicio while de ajuste da tela horario
switch(flag_tela)
{
    case 0:
        leitura = analogRead(A0);
        if((leitura > 140)&&(leitura <=150)){
            //=== Inicio do Debounce da chave up === //
            while((leitura > 140)&&(leitura <=150)){leitura = analogRead(A0);}
            delay(5);// Debounce da chave
            //=== Fim do Debounce da chave up === //
            hora++;
            if(hora >23) hora=23;
        }else if((leitura >300)&&(leitura <= 340)){
            //=== Inicio do Debounce da chave up === //
            while((leitura >300)&&(leitura <= 340)){leitura = analogRead(A0);}
            delay(5);// Debounce da chave
            //=== Fim do Debounce da chave up === //
            hora--;
            if(hora <=0) hora=0;
        }else if (leitura <= 20){
            //=== Inicio do Debounce da chave up === //
            while(leitura <= 20){leitura = analogRead(A0);}
            delay(5);// Debounce da chave
            //=== Fim do Debounce da chave up === //
            flag_tela++;
        }
        // imprime a tela piscando o campo hora
        lcd.setCursor(0,1);
        lcd.print(" ");
        lcd.setCursor(0,1);
        lcd.print(" ");
        lcd.print(":");
        if (minuto < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
        lcd.print(minuto,DEC); //imprime minutos com dois digitos
        lcd.print(":");
        if (segundo < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10

```

```

        lcd.print(segundo,DEC);//imprime segundos com dois digitos
break;
case 1:
    leitura = analogRead(A0);
    if((leitura > 140)&&(leitura <=150)){
        //=== Início do Debounce da chave up === //
        while((leitura > 140)&&(leitura <=150)){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        minuto++;
        if(minuto >59) minuto=59;
        }else if((leitura >300)&&(leitura <= 340)){
        //=== Início do Debounce da chave up === //
        while((leitura >300)&&(leitura <= 340)){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        delay(10);// Debounce da chave
        minuto--;
        if(minuto <=0) minuto=0;
        }else if (leitura <= 20){
        //=== Início do Debounce da chave up === //
        while(leitura <= 20){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        flag_tela++;
    }
    // imprime a tela piscando o campo minuto
    lcd.setCursor(0,1);
    lcd.print("      ");
    lcd.setCursor(0,1);
    if (hora < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
    lcd.print(hora,DEC);//imprime horas com dois digitos
    lcd.print(":");
    lcd.print(" ");
    lcd.print(":");
    if (segundo < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
    lcd.print(segundo,DEC);//imprime segundos com dois digitos
break;
case 2:
    leitura = analogRead(A0);
    if((leitura > 140)&&(leitura <=150)){
        //=== Início do Debounce da chave up === //
        while((leitura > 140)&&(leitura <=150)){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        segundo++;
        if(segundo >59) segundo=59;
        }else if((leitura >300)&&(leitura <= 340)){
        //=== Início do Debounce da chave up === //
        while((leitura >300)&&(leitura <= 340)){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        delay(40);
        segundo--;
        if(segundo <0) segundo=0;
        }else if (leitura <= 20){
        //=== Início do Debounce da chave up === //
        while(leitura <= 20){leitura = analogRead(A0);}
        delay(5);// Debounce da chave
        //=== Fim do Debounce da chave up === //
        flag_tela=0;
    }
    lcd.setCursor(0,1);
    lcd.print("      ");
    lcd.setCursor(0,1);
    if (hora < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
    lcd.print(hora,DEC);//imprime horas com dois digitos
    lcd.print(":");
    if (minuto < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
    lcd.print(minuto,DEC);//imprime minutos com dois digitos
    lcd.print(":");
    lcd.print(" ");
break;
}
    delay(50);
    lcd.setCursor(0,1);

```

```

    lcd.print("      ");
    lcd.setCursor(0,1);
    if (hora < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
    lcd.print(hora,DEC); //imprime horas com dois digitos
    lcd.print(":");
    if (minuto < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
    lcd.print(minuto,DEC); //imprime minutos com dois digitos
    lcd.print(":");
    if (segundo < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
    lcd.print(segundo,DEC); //imprime segundos com dois digitos
    delay(50);

    //Condição para sair da tela de ajuste pressionando o botao Select
    if((leitura >700)&&(leitura <= 750)){
        while((leitura >700)&&(leitura <= 750)){leitura = analogRead(A0);}
        delay(5); // Debounce da chave
        flag_tela = 3;
    }
    //Lógica do Timer para ajuste de tela
    tempo_atual = millis();
    if((tempo_atual - tempo_anterior)>=tempo_tela){
        flag_tela = 3;
    }

} //Fim while de ajuste da tela horário

} //Fim Ij ajustar RTC inicial

data.Day = dia;
data.Month = mes;
data.Year = CalendarYrToTm(ano);
data.Hour = hora;
data.Minute = minuto;
data.Second = segundo;
rtc.write(data);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Escolha o modo");
lcd.setCursor(0,1);
lcd.print("de operacao");
delay(2000);

//===== Fim da Tela para configuração do horário do RTC
=====

//===== Tela para seleção do modo Logger ou de decisão
=====//
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Qual operacao?");
    lcd.setCursor(0,1);
    tempo_anterior = millis();
    flag_tela = 0; //Zera a contagem de estado da Tela
    tela_modos_operacao = 0;
    while(flag_tela <1)
    { //Inicio while de seleção para entrar na tela de modo de operação
        tempo_atual = millis();
        leitura = analogRead(A0);
        if((leitura > 140)&&(leitura <=150)) //Verifica a leitura do botao Up
        {
            //=== Início do Debounce da chave up === //
            while((leitura > 140)&&(leitura <=150)){leitura = analogRead(A0);}
            delay(10); // Debounce da chave
            //=== Fim do Debounce da chave up === //
            tela_modos_operacao++;
            flag_tela = 0;
        }
        else if((leitura >700)&&(leitura <= 750)) //Verifica a leitura do botão Select
        {
            //=== Início do Debounce da chave up === //
            while((leitura >700)&&(leitura <= 750)){leitura = analogRead(A0);}
            delay(10); // Debounce da chave
            //=== Fim do Debounce da chave up === //
            lcd.clear();
        }
    }

```

```

        lcd.setCursor(0,0);
        flag_tela = 1;
    }
    if(tela_modos_operacao >1){tela_modos_operacao = 0;}

    delay(50);
    lcd.setCursor(0,1);
    lcd.print("          ");
    lcd.setCursor(0,1);
    switch(tela_modos_operacao)
    {
        case 1:
            lcd.print("LOGGER");
            break;
        case 0:
            lcd.print("TOMADA DECISAO");
            break;
        default:
            break;
    }
    if((tempo_atual - tempo_anterior)>=tempo_tela)
        {//Lógica do Timer para ajuste de tela
        tela_modos_operacao=1;
        flag_tela = 2;
        }
    }//Fim while de seleção para entrar na tela de modo de operação

//=====Verifica o início do cartão SD=====//
    lcd.clear();
    if (!sd.begin(cartao_cs)) {
        lcd.print("SD fora");
    }else{
        lcd.print(" SD inserido");
    }
    delay (1000);
//=====Fim da verificação do cartão SD=====//

    lcd.clear();
    lcd.setCursor(0,0);

}//=====Fim void
setup()=====//

void loop()
{
//=====Início do Void
Loop=====

    if(tela_modos_operacao ==1)
        {
//Início do modo de logger de dados
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("          ");
        lcd.print("Teste logger");

        rtc.read(data); //Le o rtc antes de entrar no while
        hora = data.Hour;
        minuto = data.Minute;
        horario_anterior = ((hora)*60)+(minuto); //Converte a leitura do RTC de horas e
minutos em apenas um valor contado dos minutos
        diferenca_horario = 0;
        energia_antes = millis(); //Zera o medidor de energia no in
            while(diferenca_horario <= janela_amostragem){
                rtc.read(data);
                hora = data.Hour;
                minuto = data.Minute;
                horario_atual = ((hora)*60)+(minuto); //Converte a leitura do RTC de horas e
minutos em apenas um valor contado dos minutos para a leitura mais recente
                diferenca_horario = horario_atual - horario_anterior; //Verifica a janela de
tempo entre o horário atual e o horário anterior
                emon1.calcVI(20,2000); //utiliza 20 ciclos de 60Hz para calcular a potência
real (ativa)
                potencia_real = emon1.realPower/1000; //determina a potencia Real para a
janela de 20 ciclos
                energia = energia + (millis()-energia_antes)*(potencia_real*3.333); //
determina a energia consumida em ws na janela de 20 ciclos
                ler_RTC_imprimir_tela();
            }
        }
}

```

```

        energia_antes = millis();
    }
    energia = energia*0.00000028; //converte de Ws para kWh
    rtc.read(data);
    hora = data.Hour;
    minuto = data.Minute;
    dia = data.Day;
    mes = data.Month;
    ano = tmYearToCalendar(data.Year);

//=====Anota medidas de energia no cartao sd=====//
    sd.begin(cartao_cs);
    File arquivo = sd.open("LOG.txt",FILE_WRITE); //Abre o arquivo de log de dados
    arquivo.println(" ");
    arquivo.print(dia);
    arquivo.print("-");
    arquivo.print(mes);
    arquivo.print("-");
    arquivo.print(ano);
    arquivo.print(",");
    arquivo.print(hora);
    arquivo.print(":");
    arquivo.print(minuto);
    arquivo.print(",");
    arquivo.print(energia);
    arquivo.print(";");
    arquivo.println(" ");
    arquivo.close();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("          ");
    lcd.print("Pronto!!");
    delay(2000);

//=====Anota medidas de energia no cartao sd=====//

    }//Fim do modo de logger de dados
else if(tela_modos_operacao ==0)
{ //Inicio do modo de ajuste e leitura do consumo
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("          ");
    lcd.setCursor(0,0);
    lcd.print("Modo Decisao");
    delay(1000);
} //fim do modo de ajuste e leitura do consumo

} //=====Fim do void
loop=====

//===== Função para imprimir o horário na tela do LCD
=====//
tmElements_t ler_RTC_imprimir_tela(){
    tmElements_t data;
    RTC.read(data);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("          ");
    lcd.setCursor(0,0);
    lcd.print("Data e hora");
    lcd.setCursor(0,1);
    if (data.Day < 10){ lcd.print("0");} //Imprime zero se hora forem menores do que 10
    lcd.print(data.Day);
    lcd.print("/");
    if (data.Month < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
    lcd.print(data.Month);
    lcd.print("/");
    lcd.print(tmYearToCalendar(data.Year)-2000);
    lcd.print(" ");
    if (data.Hour < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
    lcd.print(data.Hour,DEC);
    lcd.print(":");
    if (data.Minute < 10) lcd.print("0"); //Imprime zero se hora forem menores do que 10
    lcd.print(data.Minute,DEC);

```

```
    return data;  
}
```