



UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MESTRADO ACADÊMICO EM ENGENHARIA ELÉTRICA

ARTHUR NUNES FERREIRA CORREIA

**APLICAÇÕES DE SISTEMA DE MONITORAMENTO DE CONSUMO DE
ENERGIA ELÉTRICA PARA UNIDADES CONSUMIDORAS CONECTADAS EM
MÉDIA TENSÃO**

FORTALEZA

2019

ARTHUR NUNES FERREIRA CORREIA

APLICAÇÕES DE SISTEMA DE MONITORAMENTO DE CONSUMO DE ENERGIA
ELÉTRICA PARA UNIDADES CONSUMIDORAS CONECTADAS EM MÉDIA
TENSÃO

Proposta de dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará, como requisito para a obtenção do título de mestre em Engenharia Elétrica.

Orientador: Prof. PhD. Fernando Luiz Marcelo Antunes

FORTALEZA
2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C847a Correia, Arthur Nunes Ferreira.
APLICAÇÕES DE SISTEMA DE MONITORAMENTO DE CONSUMO DE ENERGIA ELÉTRICA
PARA UNIDADES CONSUMIDORAS CONECTADAS EM MÉDIA TENSÃO / Arthur Nunes Ferreira
Correia. – 2019.
93 f.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-
Graduação em Engenharia Elétrica, Fortaleza, 2019.
Orientação: Prof. Dr. Fernando Luiz Marcelo Antunes.

1. Monitoramento de energia. 2. Eficiência Energética. 3. Internet das Coisas. 4. Cadeias de Markov.
5. Criptomoedas. I. Título.

CDD 621.3

RESUMO

No Brasil, há aproximadamente 82 milhões de unidades consumidoras. Apesar de as unidades conectadas média tensão representarem apenas 0,2% do número total de unidades, seu consumo responde por aproximadamente 25% do total. Além disso, tais unidades apresentam uma maior complexidade de faturamento. Portanto, o grande volume de energia elétrica consumida por essas unidades, bem como os diversos fatores incidentes na sua tarifação podem provocar desperdícios financeiros e energéticos. Tendo em vista esses pontos, este trabalho apresenta um sistema de monitoramento de consumo de energia para unidades conectadas em média tensão, de modo a promover ao usuário um maior conhecimento a respeito do seu consumo de eletricidade e proporcionar, possivelmente, uma redução no valor mensal gasto.

O sistema desenvolvido adquire dados diretamente do medidor eletrônico utilizado pela companhia distribuidora local, descartando a necessidade de sensores e tornando o sistema com precisão ideal. Além disso, reduz consideravelmente o custo de seu desenvolvimento. Um microcontrolador ESP32 é utilizado para aquisição e envio dos dados, e uma aplicação em *Ruby* foi desenvolvida para tratamento e visualização dos dados. O sistema foi instalado na Fundação de Teleducação do Ceará, e proporcionou informações que possibilitaram uma redução no valor gasto de energia de aproximadamente 10%. Outras aplicações do sistema de monitoramento são apresentadas. Primeiramente, é feita uma análise dos dados para a geração de modelos de perfis de carga utilizando cadeias de Markov. Em seguida, um sistema de comercialização de energia utilizando a criptomoeda IOTA é analisado e implementado.

Palavras-chave: Monitoramento de energia. Eficiência Energética. Internet das Coisas. Cadeias de Markov. Criptomoedas. IOTA.

ABSTRACT

In Brazil, there are approximately 82 million consumers connected in low and medium voltage grid, consuming approximately 320 thousand GWh annually, according to ANEEL. Although the connected medium voltage units represent only 0.2% of the total number of units, their total consumption represents approximately 25% of total consumed. Such units have greater billing complexity. Therefore, the large volume of electric energy consumed by these units, as well as the various factors involved in their charging can cause financial and energy waste. Considering these points, this work presents an energy consumption monitoring system for units connected in medium voltage, in order to promote to the system user a greater knowledge about its consumption of electricity and possibly to provide a reduction in the amount spent monthly.

The developed system acquires data directly from the electronic meter used by the local distribution system operator, dismissing the need for sensors and making the system with optimum accuracy. In addition, it considerably reduces the cost of its development. An ESP32 microcontroller is used to acquire and send data, and a Ruby application was developed for data processing and visualization. The system was installed in the Teleducation Foundation of Ceará, and provided a reduction in the amount spent of energy bill of approximately 10%.

Other applications of the monitoring system are presented. Firstly, a data analysis is performed for the generation of load profile models using Markov chains. Then an energy trading system using the IOTA cryptocurrency is analyzed and implemented.

Keywords: Energy Monitoring. Energy Efficiency. Internet of Things. Markov Chains. Cryptocurrencies. IOTA

LISTA DE FIGURAS

Figura 1– IDH x Consumo de eletricidade (kWh) per capita em 124 países	13
Figura 2 – Comparativo entre a tarifa branca e a tarifa convencional.....	18
Figura 3 – Curva de carga do SIN (MW) em 05 de junho de 2019.....	22
Figura 4 – Características de carga e geração no dia 07 de maio de 2019, as 11:09.....	23
Figura 5 – Curva de geração (MW) de energia eólica no Nordeste em 05 de junho de 2019..	23
Figura 6– Funcionamento da tecnologia LoRaWAN para envio dos dados	25
Figura 7 – Diagrama de bloco funcional do ESP32	33
Figura 8 – Exemplo de requisição GET	34
Figura 9 – Arquitetura MVC do RoR.....	36
Figura 10 – Dados iniciais extraídos do sistema	37
Figura 11– Consumo de energia ativa em 12 de maio de 2019.....	38
Figura 12 – Consumo de energia ativa no mês de abril de 2019.....	38
Figura 13– Consumo de energia ativa em 2019	39
Figura 14– Demanda ativa em 3 de maio de 2019	39
Figura 15– Fator de potência no dia 10 de maio de 2019.....	40
Figura 16 - Consumo de energia ativa no dia 10 de maio de 2019	41
Figura 17 - Consumo de energia reativa no dia 10 de maio de 2019	41
Figura 18– Análise de demanda contratada no período de dezembro de 2018.....	44
Figura 19– Representação de uma cadeia de Markov	46
Figura 20– Exemplo de uma matriz de transição de uma cadeia de Markov	47
Figura 21– Representação do processo de modelagem	48
Figura 22 – Perfil de consumo obtido através de simulação com MATLAB	49
Figura 23 – Fluxograma da modelagem desenvolvida.....	50
Figura 24 - Média dos valores obtidos com sistema de monitoramento (azul) e simulação (vermelho)	51
Figura 25 - Comparação de transições de estado entre dados reais (esquerda) e dados simulados (direita)	52
Figura 26 - Histograma de ocorrências de estado com dados reais (esquerda) e dados simulados (direita)	52
Figura 27 – Princípio de funcionamento da rede <i>Tangle</i>	56
Figura 28 – Sistema proposto	58

Figura 29 - Fluxograma do processo de comercialização de energia desenvolvido	59
Figura 30 – Quantidade de energia disponível para a venda.....	60
Figura 31 – Energia disponível para compra.....	60
Figura 32 – Página de finalização da oferta	61
Figura 33 – Comparação de balanço das carteiras de IOTA	62

LISTA DE TABELAS

Tabela 1 – Comparação entre tecnologias de envio de dados sem fio	26
Tabela 2 – Comparação entre sistemas de monitoramento de parâmetros elétricos	27
Tabela 3 – Limites de erro percentual para medidores de energia ativa	30
Tabela 4 – Limite de erro percentual para medidores de energia reativa.....	31

LISTA DE ABREVIATURAS E SIGLAS

CRC	<i>Cyclic Redundancy Check</i>
DAG	<i>Directed Acyclic Graph</i>
DMCR	Demanda Máxima Corrigida Registrada
FUNCAP	Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico
GPRS	General Packet Radio Services
HTTP	<i>Hypertext Transfer Protocol</i>
IOT	<i>Internet of Things</i>
IRI	<i>IOTA Reference Implementation</i>
LSB	<i>Less Significant Bit</i>
M2M	<i>Machine-to-Machine</i>
MBPS	Megabits por Segundo
MEC	Ministério da Educação
MSB	<i>Most Significant Bit</i>
NBR	Norma Brasileira
PLC	<i>Programmable Logic Controller</i>
SEINFRA	Secretaria de Infraestrutura do Estado do Ceará
SPI	<i>Serial Peripheral Interface</i>
SSID	<i>Service Set Identifier</i>
TC	Transformador de Corrente
TP	Transformador de Potencial
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
UFER	Unidade para Faturamento de Energia Reativa
VPS	<i>Virtual Private Server</i>
WMM	Wi-Fi Multimedia

SUMÁRIO

1 INTRODUÇÃO	12
2 SISTEMAS DE MONITORAMENTO	15
2.1 Medição inteligente	15
2.2 Eficiência energética	16
2.3 Tarifação.....	16
2.3.1 Tarifação de unidades do grupo B	17
2.3.2 Tarifação de unidades do grupo A	18
2.3.2.1 Demanda.....	18
2.3.2.2 Energia Reativa e Fator de Potência	19
2.3.2.3 Modalidade tarifária	21
2.4 Sistemas de monitoramento	22
2.4.1 Sensores	24
2.4.2 Conversores analógico-digitais.....	24
2.4.3 Hardware de processamento de dados.....	25
2.4.4 Software de processamento e visualização dos dados.....	26
2.5 Conclusões	27
3 SISTEMA PROPOSTO	29
3.1 Conceito	29
3.1.1 Medidores eletrônicos de energia	30
3.1.2 NBR 14522:2008	31
3.2 Arquitetura de desenvolvimento	32
3.2.1 Hardware de aquisição dos dados.....	32
3.2.2 Software de desenvolvimento	35
3.3 Resultados obtidos	43
3.4 Conclusão parcial.....	44
4 POSSÍVEIS APLICAÇÕES DO SISTEMA DE MONITORAMENTO	45
4.1 Modelagem de perfil de consumo	45
4.1.1 Cadeias de Markov.....	45
4.1.1.1 Modelagem	47
4.2 Transação de energia utilizando criptomoeda	53
4.2.1 Introdução	53
4.2.2 IOTA	55

4.2.3 Sistema proposto.....	57
4.2.4 Resultados.....	59
4.5 Conclusão.....	67
5 CONCLUSÕES PARCIAIS	67
5.1 Atividades futuras.....	68
REFERÊNCIAS	69
ANEXOS	72
APÊNDICE	74
Apêndice A.....	74
Apêndice B.....	79
Apêndice C – Artigo ISIE	93

1 INTRODUÇÃO

A energia elétrica possui um papel essencial no desenvolvimento e bem-estar de uma sociedade. Há registros antigos do conhecimento de sua existência, tendo estudos efetivos de sua aplicabilidade a partir do desenvolvimento da primeira máquina eletrostática pelo físico alemão Otto von Guericke, passando pelas descobertas de Benjamin Franklin e pela “guerra das correntes”, entre George Westinghouse e Nikola Tesla, que defendiam a utilização da corrente alternada na distribuição da energia elétrica, e Thomas Edison, que defendia o uso da corrente contínua, até os dias atuais, onde é utilizada em aplicações que elevam a produtividade e o conforto da população, como é o caso da Internet e refrigeradores de ar.

Quedraogo (2013) investigou a relação entre o consumo de eletricidade e o IDH em quinze países em desenvolvimento, entre 1988 e 2008. No curto prazo, não houve uma correlação significativa entre o consumo de eletricidade per capita e o índice, mas no longo prazo, um aumento em 1% no consumo de eletricidade promove um aumento no IDH de 0,22%. Além disso, o aumento de 1% no preço da energia resulta em um impacto negativo 0,11%.

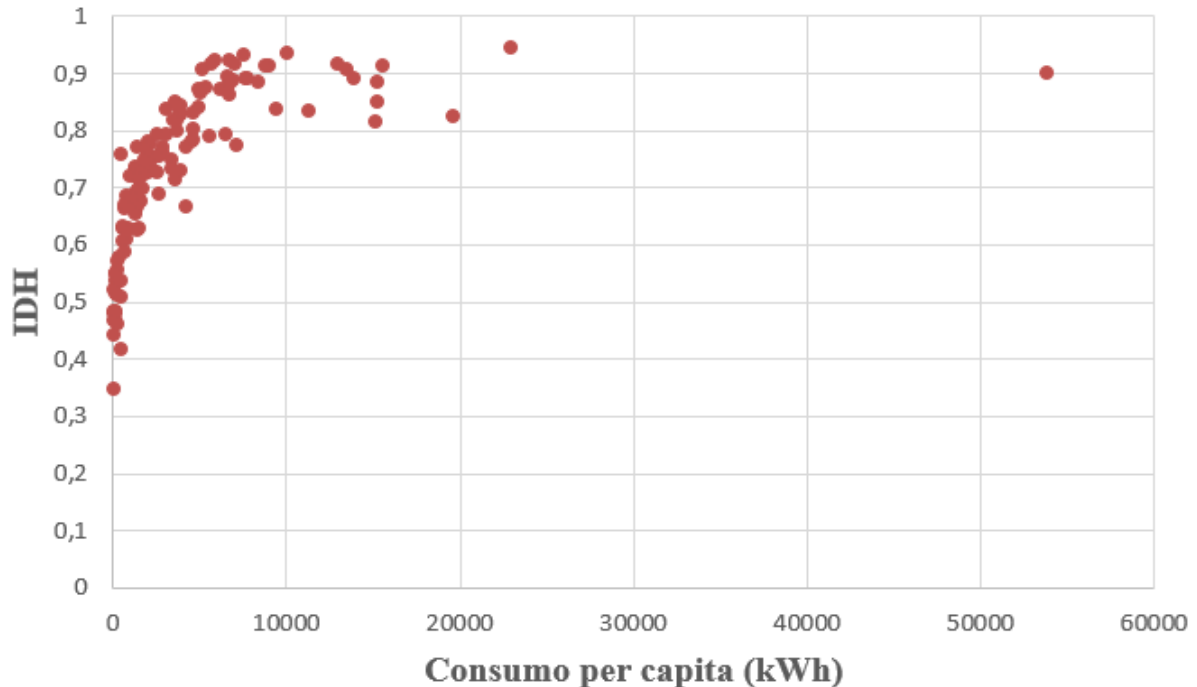
Niu et. al. (2013) afirma que cinco são as principais contribuições da energia elétrica para a satisfação da população, dentre eles melhor conservação de alimentos, drogas e vacinas, contribuição da iluminação e da tecnologia da informação para o aumento da produtividade, utilização para o aumento do bem-estar, como é o caso de aquecedores e refrigeradores, seu uso dentro de residências é menos nocivos comparando-se a outras fontes de energia, como a queima do carvão, e a eletricidade promove um aumento de oportunidades de desenvolvimento da população.

Na Figura 1 está demonstrado graficamente a relação entre o IDH e o consumo de eletricidade per capita (kWh), em uma análise contendo dados de 124 países. O país com maior consumo de eletricidade per capita dentre os listados é a Islândia, com um consumo de 53.800 kWh per capita e possui um IDH de 0,899. O país com maior IDH é a Noruega, com um valor de 0,944 e possuindo um consumo per capita de eletricidade de aproximadamente 23.000 kWh. Do outro lado dessa perspectiva, Haiti é o país com menor consumo, com aproximadamente 39 kWh per capita e um IDH de 0,483. Níger possui o pior IDH dentre os países listados, com um valor de 0,348 e um consumo per capita de 51,4 kWh. Com bases em tais dados, e analisando a Figura 1, é notória a importância do consumo de energia elétrica para o desenvolvimento de uma região.

Além disso, o preço da energia elétrica no Brasil vem crescendo de forma considerável nos últimos anos. Por ter sua matriz essencialmente constituída por geração hidrelétrica, as

secas ocorridas nos últimos anos resultaram em um aumento generalizado no preço pago pelo kWh.

Figura 1– IDH x Consumo de eletricidade (kWh) per capita em 124 países



Fonte: Próprio autor, com dados de <www.worldbank.org>

Tendo em vista a essencialidade da energia elétrica para o desenvolvimento de uma sociedade, e os aumentos recentes no seu preço, se faz útil um maior questionamento e crítica a respeito do que se é consumido. Uma maior necessidade ocorre aos clientes conectados em média e alta tensão, devido, principalmente, ao volume de energia consumido e aos altos valores pagos em contas de energia. Segundo a ANEEL, em 2018, esses clientes representavam 0,2% do total de unidades consumidoras de energia elétrica no Brasil, e respondiam por aproximadamente 27% de todo o consumo. Além disso, tais clientes pagam por diversos fatores a respeito da utilização da energia elétrica, como energia ativa e demanda, sendo necessária uma análise crítica mais aprimorada a respeito de seu uso.

Portanto, tendo em vista tais observações, o objeto deste trabalho é o desenvolvimento de um sistema capaz de monitorar de forma inteligente o consumo de energia elétrica em unidades de média tensão, mostrando possibilidades para o usuário obter um consumo mais eficiente e uma possível redução na conta de energia.

O artigo desenvolvido a partir deste projeto foi publicado no 28º IEEE *International Symposium on Industrial Electronics (ISIE 2019)*, Vancouver/Canadá, sob o título “*Design of real-time energy management system for medium voltage grid-connected customer*”, e também

no ISES *Solar World Congress* (SWC 2019), Santiago/Chile, sob o título “*Development of an Energy Commercialization Distributed Generation using IOTA*”. Os artigos podem ser encontrados nos apêndices deste trabalho.

Os seguintes capítulos deste trabalho estão da seguinte forma estruturados: No Capítulo 2 encontra-se uma revisão teórica a respeito de medidores inteligentes de energia. No Capítulo 3 detalha-se a respeito do sistema de monitoramento desenvolvido e suas possíveis aplicações. No Capítulo 4, são feitas análises e resultados obtidos com a utilização do sistema de monitoramento utilizado. O Capítulo 5 apresenta as conclusões deste trabalho.

2 SISTEMAS DE MONITORAMENTO

Diversos fatores são capazes de influenciar o fluxo de energia elétrica, desde sua produção até o seu consumo. A necessidade constante de sua utilização, o impacto negativo causado por sua ausência, os custos envolvidos e o risco à segurança dos que trabalham diretamente com ela ou que a utilizam são os principais motivadores para o desenvolvimento de sistemas de monitoramento de energia elétrica.

Nesse capítulo, são abordados conceitos essenciais para o entendimento da necessidade desse tipo de sistema e uma revisão sobre algumas tipologias para diversas funcionalidades será realizada.

2.1 Medição inteligente

Barai (2015) define medição inteligente como dispositivos de medição capazes de gravar a quantidade de potência utilizada, e enviar os dados automaticamente para um sistema de gerenciamento, alterando a operação da rede elétrica e promovendo seu uso de forma mais segura. Dentro dessa definição, alguns conceitos são desenvolvidos, como sensoriamento, inteligência computacional no *hardware* de medição, protocolos de comunicação, tecnologia de transmissão dos dados, segurança da informação, e a possibilidade de utilizar-se um método de comunicação de duas vias, onde não só seja realizada a leituras dos parâmetros adquiridos, como possa realizar também comandos para a atuação do dispositivo.

O estudo da medição inteligente baseia-se nas mais diversas motivações. Barai (2015) aponta, dentre diversas vantagens, a rápida detecção de faltas de energia pelo responsável pelo fornecimento, aumento do conhecimento do consumidor, passando a conhecer melhor o seu perfil de consumo e possíveis problemas e melhor planejamento e gerenciamento da rede. Hong (2010) utiliza medição inteligente para calcular perdas em linhas de distribuição. Henderson (2011) e Visilatchi (2017) utilizam sistemas inteligentes de medição para controle e prevenção de furto de energia. Gross (2014) e Hong (2014) utilizam dados fornecidos a partir de sistemas de medição inteligentes para utilizar algoritmos de processos estocásticos para previsão do consumo de energia.

2.2 Eficiência energética

Segundo a EPE, eficiência energética significa gerar a mesma quantidade de energia com menos recursos naturais ou obter o mesmo trabalho realizado com menos energia. Além disso, faz parte do processo do consumo eficiente de energia evitar desperdícios, evitando a realização de trabalhos desnecessários.

Do ponto de vista das empresas do setor elétrico, um consumo eficiente de energia promove uma redução de perdas no processo de transmissão e distribuição e uma prevenção ou postergação de investimentos na rede elétrica. Para o consumidor final, eficiência energética promove menos gastos e mais segurança.

O Programa Nacional de Conservação de Energia Elétrica (PROCEL), sob a responsabilidade do Ministério de Minas e Energias e coordenado pela Eletrobrás, visa promover o consumo eficiente de energia e combater o seu desperdício. O programa trabalha qualificando aparelhos eletroeletrônicos por modo de consumo, no apoio às prefeituras no planejamento e implantação de projetos de iluminação pública, treinamentos para indústrias e outros consumidores, entre outras ações.

A eficiência energética pode provir não somente da redução do consumo de energia. No Brasil, os clientes de baixa tensão podem ser tarifados pela tarifa branca, onde o preço da energia varia de acordo com a hora do dia. Portanto, um conhecimento por parte do cliente de seu perfil de consumo e carga podem promover um consumo mais eficiente de energia. O mesmo ocorre com clientes conectados em média e alta tensão. Nesses casos, os consumidores podem ser tarifados dentro de três possibilidades: verde, azul e optante pelo grupo B. Cada modo de tarifação contém suas particularidades, e, novamente, um conhecimento das características de consumo podem resultar numa eficiência energética maior. Mais detalhes a respeito do modo de tarifação dos clientes serão abordados no próximo tópico.

2.3 Tarifação

No Brasil, os consumidores de energia elétrica podem ser tarifados de diversas formas. Além disso, algumas tarifas e multas podem incidir na conta final do consumidor. Portanto, um maior conhecimento a respeito do que é pago na fatura de energia pode promover uma redução de custos.

2.3.1 Tarifação de unidades do grupo B

Unidades do grupo B são aquelas conectadas à rede de baixa tensão. Estas são tarifadas de forma monômnia, ou seja, é cobrada apenas pela quantidade de energia elétrica ativa consumida. Portanto, para a obtenção de uma redução no valor pago na conta de energia, o cliente deve diminuir o tempo de utilização de aparelhos conectados à rede elétrica ou optar por aparelhos mais eficientes.

Existem dois modos de tarifação para unidades conectadas em baixa tensão: convencional e a tarifa branca. No modo convencional, a unidade para o mesmo preço de energia em qualquer horário do dia, ou em qualquer dia da semana.

A tarifa branca proporciona ao consumidor pagar diferentes valores em função da hora e do dia da semana em que a respectiva energia foi consumida. O modelo branco de tarifação pode ser adotado por unidades consumidoras conectadas em baixa tensão de todos os níveis (127, 220, 380 e 440 V), em novas conexões ou já existentes que possuem uma média de consumo maior que 250 kWh por mês.

O modelo branco de tarifação possui três níveis de valores para a energia consumida:

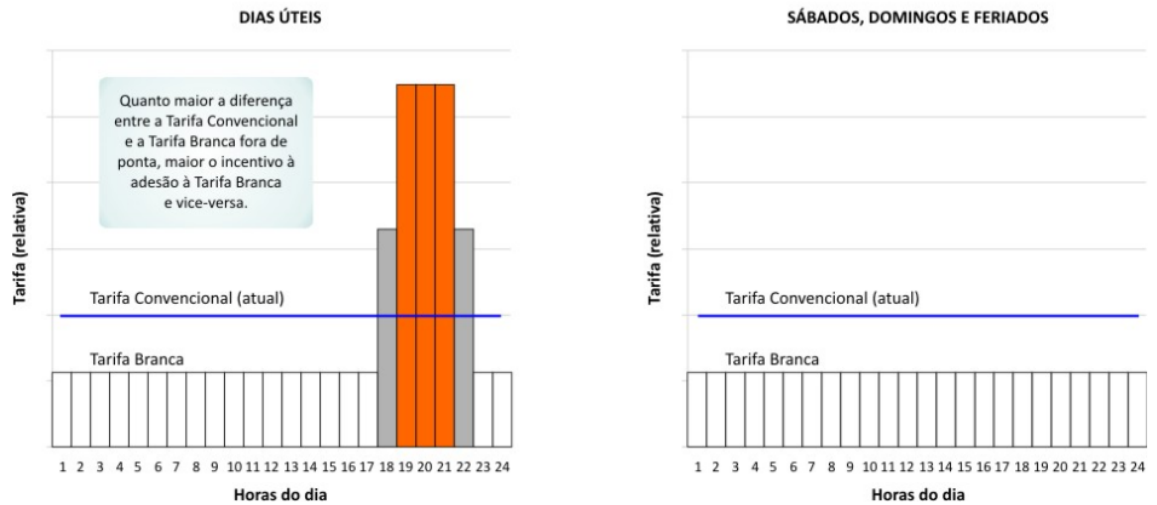
- Ponta, período em que a rede elétrica se encontra mais sobrecarregada, e, conseqüentemente, o valor da energia é mais caro;
- Intermediário, com valor homônimo e normalmente uma hora posterior e anterior ao período de ponta;
- Fora de ponta, período onde a rede elétrica está menos sobrecarregada, resultando em um valor mais barato de energia;

Os valores intermediários e de ponta somente incidem em dias úteis. Nos finais de semana e feriados, somente os valores fora de ponta são contabilizados.

Na Figura 2 consta um gráfico comparativo entre a tarifa convencional e branca. Na Figura 2, os períodos em laranja são os de ponta, os que estão pintados em cor cinza são os períodos intermediários, e os períodos em branco são fora de ponta. Segundo a ANEEL, os respectivos horários são homologados pela ANEEL nas revisões tarifárias que ocorrem em média a cada 5 anos.

A tarifa branca visa um incentivo financeiro para que os consumidores de baixa tensão transfiram o consumo de energia elétrica de horários onde a rede elétrica se encontra mais congestionada para horários mais livres. Como pode ser visto na imagem, a tarifa nos horários de fora de ponta são mais econômicos que a tarifa convencional, portanto, pode possibilitar uma redução significativa no valor de energia pago pelo consumidor.

Figura 2 – Comparativo entre a tarifa branca e a tarifa convencional



Fonte: <www.aneel.gov.br/tarifa-branca>

2.3.2 Tarifação de unidades do grupo A

Unidades do grupo A são aquelas conectadas às redes de médias e alta tensão. Alguns fatores devem ser considerados na tarifação em unidades do grupo A.

2.3.2.1 Demanda

As unidades conectadas às redes de média e alta tensão são tarifadas de forma binômia (com exceção de alguns casos que podem optar serem tarifados como unidades de baixa tensão), ou seja, além da energia ativa, os consumidores pagam também por um valor de demanda. Tal valor é cobrado de modo a auxiliar as distribuidoras de energia no planejamento de manutenção e expansão de sua rede elétrica. A demanda registrada da unidade é a potência média em intervalos de 15 minutos. Portanto, a cada dia, 96 registros de demanda são registrados em cada unidade consumidora.

O valor da demanda contratada por impactar de forma significativa o valor da conta de energia. Normalmente, o usuário da rede contrata um valor fixo, pago por unidade de kW. O valor a ser pago pela demanda é o menor entre o valor registrado e a demanda contratada. Portanto, mesmo que a unidade não registre o seu valor contratado, ainda assim este será o valor a ser pago. No caso contrário, caso ultrapasse ao menos uma vez o valor da demanda contratada, será cobrado ao cliente uma multa, que custa o dobro do valor pago por unidade de kW para cada unidade ultrapassada, além do valor registrado normalmente.

As demandas são registradas em dois horários: ponta e fora de ponta.

- Ponta: Período definido pela concessionária de energia de três horas de duração consecutivas, contabilizadas em dias úteis, não sendo consideradas em finais de semana e feriados nacionais;
- Fora de ponta: Período composto pelas demais horas do dia complementares ao horário de ponta, e todos os horários dos finais de semana e feriados nacionais;

A forma como a demanda é cobrada varia conforme as modalidades tarifárias, que serão apresentadas futuramente neste trabalho.

2.3.2.2 Energia Reativa e Fator de Potência

O conceito de fator de potência foi inicialmente estabelecido por (BUCHHOLZ, 1922), afirmando que tal grandeza representa a razão entre a energia entregue à carga e a máxima energia que poderia ser entregue à mesma, mantida as perdas na linha de alimentação. Tal definição continua válida até hoje. Em resumo, se comparando duas cargas com mesmo valor de potência ativa, aquela com menor FP necessita de uma maior quantidade de energia entregue para que realize o mesmo trabalho da outra carga. Pela definição da ANEEL, fator de potência pode ser definido conforme Eq. (1).

$$fp = \frac{E_P}{\sqrt{E_P^2 + E_Q^2}} \quad (1)$$

Onde E_P e E_Q representam, respectivamente, as energias elétricas ativa e reativa consumidas em determinado tempo.

Existem diversos conceitos que definem o conceito de energia reativa. Entretanto, do ponto de vista de faturamento, a de maior relevância é a da ANEEL, que cita tal energia como aquela que circula entre os campos elétricos e magnéticos sem realizar trabalho, expressa em quilovolt-ampere-reativo-hora (kvarh).

A Portaria nº 1.569 do DNAEE reformulou o fator de potência de referência com limite mínimo de 0,92. Tal medida é afirmada pela Resolução Normativa nº 414/2010, pela ANEEL. A Resolução Normativa nº 414 estabelece a cobrança respectivas de consumo de energia reativa excedente e demanda de potência ativa excedente conforme as Eqs. (2) e (3), respectivamente.

$$E_{RE} = \sum_{T=1}^n \left[EEAM_T \times \left(\frac{f_R}{f_T} - 1 \right) \right] \times VR_{ERE} \quad (2)$$

$$D_{RE}(p) = \left[\text{MAX}_{T=1}^n \left(PAM_T \times \frac{f_R}{f_T} \right) - PAF(p) \right] \times VR_{DRE} \quad (3)$$

Onde:

E_{RE} = Total cobrado em (R\$) pelo consumo excedente de energia reativa;

$EEAM_T$ = Total de energia elétrica ativa medida em intervalos de uma hora;

f_R = Fator de potência de referência, com valor de 0,92;

f_T = Fator de potência no intervalo de uma hora, calculado conforme Eq. (1);

VR_{ERE} = valor da tarifa cobrado sobre energia reativa excedente, em R\$/kvarh;

$D_{RE}(p)$ = valor em cada posto tarifário, referente à demanda excedente de potência reativa;

PAM_T = demanda de potência ativa medida no intervalo de uma hora (nesse caso, igual ao valor de energia ativa consumida na respectiva hora);

$PAF(p)$ = demanda de potência ativa faturável, medido em kW;

VR_{DRE} = valor de referência pago por unidade de demanda reativa ultrapassada, em (R\$/kW);

MAX = identifica o maior valor registrado nos períodos avaliados;

T = intervalos de uma hora;

p = posto tarifário, ponta ou fora ponta;

Conforme Eqs. (2) e (3), os cálculos de cobrança de multa por consumo excedente de energia reativa é realizada de modo horário. A cada hora, calcula-se o consumo de energia ativa e energia reativa, e, conforme Eq. (1), o respectivo fator de potência do horário. Caso seja sempre maior que 0,92, não será realizada nenhuma cobrança extra com relação a esse aspecto. Entretanto, caso contrário, incidem-se taxas baseadas nas fórmulas recentemente apresentadas.

Por seis horas estabelecidos pelas distribuidoras de energia, entre 23h30 e 6h30, o fator de potência de referência deve ser de 0,92 capacitivo. Portanto, qualquer valor de fator de potência indutivo em tais horários não implica no pagamento de multa. Nos horários restantes, o consumidor deve obedecer ao valor de referência de 0,92 indutivo. Portanto, cada período horário deve apresentar fator de potência superior a 0,92 indutivo, ou qualquer valor capacitivo, para que fique isento da cobrança de multa.

Para a correção do fator de potência, deve-se inicialmente identificar a origem do problema. Caso as cargas que ocasionem um baixo fator de potência sejam lineares, a instalação de banco de capacitor se faz uma solução adequada. Entretanto, se for detectado baixo fator de

potência em decorrência da presença de harmônicos e cargas não-lineares, a instalação de capacitores implica em riscos, sendo mais recomendada a utilização de filtros de harmônicos.

2.3.2.3 Modalidade tarifária

Os consumidores de unidades conectadas em média tensão podem optar por duas opções de modalidades tarifárias: optante pelo grupo B, azul e verde.

- Optante pelo grupo B: Conforme Artigo 100 da Seção IV da RN 414/2010 da ANEEL, os clientes conectados em rede primária que possuem potência de transformação igual ou inferior a 112,5 kVA, ou aqueles que se localizam em área turística ou de veraneio podem optar por serem tarifados como unidades do grupo B, ou seja, pagando apenas um valor fixo por kWh consumido, sem a necessidade de pagar pela demanda contratada;
- Azul: Conforme Artigo 55 da Seção II da RN 414/2010 da ANEEL, o cliente optante pela modalidade tarifária azul contrata dois valores de demanda, um para o horário de ponta e outro para o horário fora de ponta, ambos com valores distintos por unidade de potência, pagos em R\$/kW, sendo maior o valor pago no horário de ponta. Para o consumo de energia, há duas tarifas distintas para energia consumida nos horários de ponta e fora de ponta, sendo o valor pago no horário de ponta o de maior valor.
- Verde: Conforme Artigo 56 da Seção II da RN 414/2010 da ANEEL, o cliente optante pela modalidade verde contrata uma tarifa única para a demanda. Portanto, para a cobrança da demanda, não difere se o valor mensurado foi registrado no horário de ponta ou fora de ponta. Entretanto, para o consumo de energia, incidem dois valores, um para o horário de ponta e outro para o horário fora de ponta, sendo o primeiro de maior valor se comparado ao segundo.

Embora a tarifa azul possua dois valores de demanda contratados, ela se torna mais vantajosa para os clientes que têm um alto consumo no horário de ponta, tendo em vista que o valor da energia no horário de ponta para os clientes optantes pela tarifa verde é consideravelmente maior que os valores para os que optam pelo modelo azul de tarifação.

Os clientes conectados em alta tensão (maior ou igual a 69 kV) somente podem optar pelo modelo de tarifação azul.

2.4 Sistemas de monitoramento

A utilização de sistemas de monitoramento em sistemas elétricos se faz presente nas mais diversas aplicações. São utilizadas no momento da geração de energia, em gerações centralizadas como em usinas de geração distribuída, em linhas de transmissão, em redes de distribuição e também no consumo. A importância de tais sistemas baseia-se, principalmente, na garantia da confiabilidade da rede, na melhoria de performance com consequente redução de perdas e no processo de faturamento de seu consumo.

Para o perfeito funcionamento da rede, diversos princípios devem ser obedecidos. Primeiramente, pela falta de uma tecnologia capaz de armazenar energia em larga escala, a geração de energia e seu respectivo consumo devem ocorrer de forma simultânea, em mesma potência. Caso contrário, o controle e o funcionamento do sistema estarão comprometidos. Para esse tipo de controle, há a necessidade de um monitoramento refinado para o controle do sistema. A ONS fornece um monitoramento em tempo real da carga e geração, divididos por subsistema (SIN, Norte, Nordeste, Sudeste / Centro-Oeste e Sul), com diferenciação por tipo de energia produzida (eólica, hidráulica, nuclear, solar e térmica), diferenciando-se também o que é consumido internamente no país ou exportado, e o que é produzido no país ou que é exportado. A Figura 3 retrata a curva de carga do SIN no dia 05 de junho de 2019.

Figura 3 – Curva de carga do SIN (MW) em 05 de junho de 2019



Fonte: <<http://ons.org.br/paginas/energia-agora/carga-e-geracao>>

A Figura 4, também extraída do site da ONS, detalha a geração e consumo no SIN por tipo de geração e origem ou destino da energia produzida no dia 07 de maio de 2019.

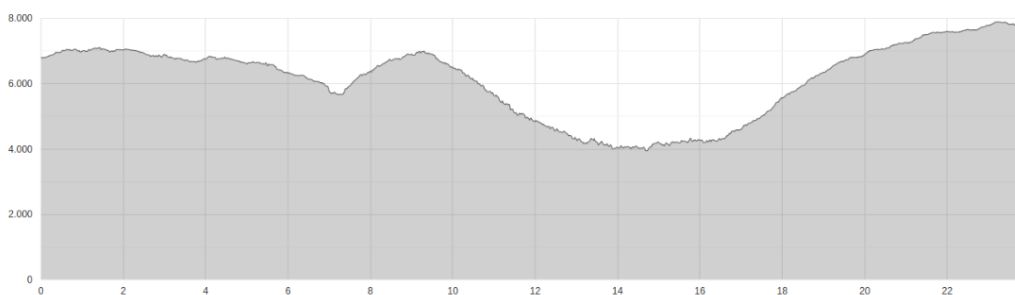
Figura 4 – Características de carga e geração no dia 07 de maio de 2019, as 11:09

07/05/2019 11:09	
Carga:	74277,2 MW
Exportação:	0,2 MW
Ger. Eólica:	2612,8 MW
Ger. Hidráulica:	63116,8 MW
Ger. Térmica:	6536,8 MW
Ger. Nuclear:	645,9 MW
Ger. Solar:	1365,0 MW
Importação:	0,0 MW

Fonte: <<http://ons.org.br/paginas/energia-agora/carga-e-geracao>>

A Figura 5 demonstra a curva de geração da energia eólica no Nordeste do Brasil no dia 05 de junho de 2019.

Figura 5 – Curva de geração (MW) de energia eólica no Nordeste em 05 de junho de 2019



Fonte: <<http://ons.org.br/paginas/energia-agora/carga-e-geracao>>

As perdas nos sistemas elétricos, por efeito Joule ou por furto e fraude de energia, também justificam a necessidade de utilização de sistemas de monitoramento em sistemas elétricos. Segundo a ABRADDEE, no ano de 2016, aproximadamente 14% de perdas foi registrado considerando todas as distribuidoras no Brasil. Wang *et al.* (2010) detalha o desenvolvimento de um sistema em tempo real de análise e gerenciamento de perdas em sistemas de potência

Sistemas de monitoramento de sistemas elétricos também são instalados dentro de unidades consumidoras com diversas finalidades. Dupont (2017) apresenta um sistema embarcado utilizando Raspberry PI para monitorar a geração e consumo de energia de um sistema de bombeamento utilizando placas fotovoltaicas. Pereira (2018) desenvolve um sistema de monitoramento com um ESP32 para monitorar condições ambientais e geração de usinas

fotovoltaicas instaladas em Fortaleza/CE, Maracanaú/CE e Colônia/Alemanha. Rajasekaran (2017) utiliza um sistema não-invasivo com Arduino e algoritmos de aprendizado de máquina para desenvolver um sistema de controle de carga em uma residência.

Alguns componentes são relevantes em um sistema de monitoramento de parâmetros elétricos, como sensores, conversores analógico-digitais, *hardware*, e *software* de tratamento e aquisição dos dados.

2.4.1 Sensores

Sensores são responsáveis pela transformação de fenômenos físicos em sinais elétricos, de modo tais grandezas possam ser analisadas por um dispositivo eletrônico. Para monitoramento de sistemas elétricos, normalmente utilizam-se sensores de tensão e corrente, pois, a partir de tais grandezas, é possível analisar uma larga variedade de dados. Apesar de grandezas elétricas, a mensuração de tensão e corrente requer a utilização de sensores para que haja adaptabilidade de parâmetros elétricos com o resto do sistema e para reduzir valores normalmente altos. TP e TC são, respectivamente, sensores de tensão e corrente comumente utilizados.

2.4.2 Conversores analógico-digitais

A utilização de conversores analógicos-digitais se faz necessário para a conversão de sinais de saída dos sensores, que normalmente são analógicos, em sinais digitais, os quais podem ser entendidos por um *hardware* de processamento de dados. É comum que o próprio *hardware* possua um conversor analógico-digital, mas em aplicações que requerem maior exatidão, pode-se utilizar circuitos integrados de *Analog Front-End*, como é o caso do MCP3913 ou ATSENSE 301. Há no mercado alguns circuitos mais robustos, que inclusive realizam o processamento inicial dos dados e fornecem uma grande quantidade de informação, como frequência, consumo ativo e reativo, valores de potência e valores eficazes, como é o caso do ATM90E32S.

Schlobohm (2005) afirma que a velocidade mínima de amostragens para uma um bom resultado de amostragem é de 16 amostras por ciclo para a determinação com classe de exatidão de 1% valores eficazes de tensão e corrente.

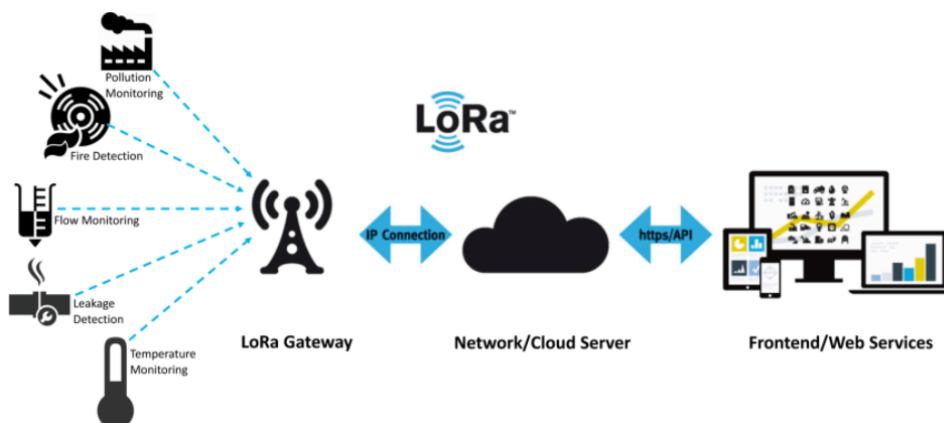
2.4.3 Hardware de processamento de dados

Utiliza-se em sistema de monitoramento de parâmetros elétricos microcontroladores para o tratamento inicial dos dados adquiridos, e posterior envio ao sistema final. É possível encontrar diversos modelos apropriados para tal finalidade. Alguns parâmetros são relevantes para a escolha de um *hardware* de processamento, como nível de tensão, SPI, UART, possibilidades de envio dos dados e velocidade de processamento.

O protocolo de envio dos dados pelo *hardware* de processamento é um fator relevante. Existem diversas possibilidades, cada uma variando entre custo, complexidade e necessidade da aplicação. Yimin (2011) utiliza GPRS para envio dos dados. A utilização desse tipo de tecnologia é recomendada quando não há presença de sinal de Internet no local, como em localizações mais remotas e que possuam sinal de telefonia. Dentre as desvantagens de utilizar o GRPS, vale ressaltar a necessidade da utilização de *chips* de dados ou telefonia para a transmissão dos dados, o que pode gerar um custo recorrente. Além disso, GPRS apresenta baixa taxa de transmissão de dados.

Outra tecnologia de transmissão que pode ser utilizada é o LoRaWAN, que é uma tecnologia recente capaz de enviar dados com grande alcance. Sahoo; Patnaik (2017) utiliza LoRaWAN para o monitoramento de sistemas de iluminação alimentados por energia solar, e afirmam que esta tecnologia tem o alcance de 12 a 15 km em zona rural e entre 3 e 5 km em zona urbana. Uma desvantagem desta tecnologia é o elevado custo se comparado a outras tecnologias, devido à necessidade da presença de um *Gateway* para o recebimento dos dados. A Figura 6 detalha o funcionamento do LoRaWAN.

Figura 6– Funcionamento da tecnologia LoRaWAN para envio dos dados



Bluetooth também é uma tecnologia que pode ser utilizada em sistemas de monitoramento. Entretanto, para que seja utilizado, há a necessidade de um segundo protocolo de envio dos dados para que o usuário à distância consiga visualizá-los, tendo em vista que o alcance de tal tecnologia é curto.

As tecnologias de transmissão de dados com fios, como RS232, também não são recomendadas para o envio dos dados ao sistema final, tendo em vista a falta de praticidade de sua utilização. Irmak (2017) apresenta a Tabela 1 demonstra uma análise comparativa entre algumas tecnologias propostas.

Tabela 1 – Comparação entre tecnologias de envio de dados sem fio

Tecnologia	ZigBee	GPRS/GSM	WiFi	Bluetooth
Área de utilização	Controle e monitoramento	Larga área de cobertura, voz e dados	Web, e-mail, vídeo	Substitui o cabo
Taxa de transmissão (kb/s)	20-250	>128	>54000	720-3000
Alcance (m)	1-1600	>1000	1-100	"1/10"
Complexidade	Simple	Complexo	Muito complexo	Complexo
Vantagens	Resistência, baixo consumo de energia	Acessibilidade	Velocidade, flexibilidade	Baixo custo, facilidade

Fonte: (IRMAK, 2017)

A Tabela 2 consta uma análise comparativa entre diversos trabalhos, considerando *hardware* de aquisição, transmissão de dados, supervisão e aplicação do trabalho.

2.4.4 Software de processamento e visualização dos dados

Todas as informações necessárias devem ser tratadas por um nível de processamento mais elevado. Comumente, *hardware* de aquisição com alto processamento requer um alto investimento, principalmente pelo fato de haver a necessidade da instalação do *hardware* em cada unidade monitorada, sendo, portanto, mais viável a utilização de sistemas como uma aplicação, um computador ou VPS, capazes de promover o uso de um processamento mais robusto e centralizado com todas as informações adquiridas.

Tabela 2 – Comparação entre sistemas de monitoramento de parâmetros elétricos

Autor	Hardware de aquisição	Transmissão de dados	Supervisão	Aplicação
SAHOO; PATNAIK, (2017)	Arduino	LoRaWAN	Simulink 11.0	Monitoramento de lanternas solares
IRMAK; KÖSE; GÖÇMEN, (2016)	ZigBee	ZigBee	MATLAB	Monitoramento de consumo de energia em <i>Smart Homes</i>
HAN <i>et al.</i> , (2014)	ZigBee	ZigBee e PLC	Desenvolvimento próprio	Monitoramento de geração e consumo de energia
MUMTAZ, (2015)	PLC	Ethernet/PLC	SCADA utilizando Visual Studio	Monitoramento de consumo de energia
YIMIN; ZHENGLI, (2011)	DTS343	GPRS	Desenvolvido por VC6.0 e Oracle	Monitoramento de consumo para indústrias
DUPONT, (2017)	Raspberry PI	WiFi	Desenvolvimento próprio utilizando PHP	Monitoramento de uma usina fotovoltaica
PEREIRA, (2018)	ESP32	WiFi	Desenvolvimento próprio	Monitoramento aplicado em plantas de microgeração fotovoltaica

Fonte: Próprio autor

Aplicações WEB, aplicativos *iOS* ou *Android* podem ser utilizados para o desenvolvimento de um sistema supervisorio para análise dos dados mensurados. O alto nível de linguagem fornecidos por tais sistemas fornecem capacidade alta de processamento e visualização das informações necessárias.

2.5 Conclusões

Neste capítulo, foram introduzidos inicialmente alguns conceitos importantes com relação a sistemas de monitoramento, como medição inteligente, eficiência energética e uma breve revisão sobre o modelo de tarifação brasileiro.

A tarifação foi explanada separando-se entre unidades conectadas em baixa tensão e unidades conectadas em média e alta tensão. Para as unidades de baixa tensão, foi visto que seu modelo de tarifação se baseia exclusivamente no consumo de energia ativa. A adoção do modo branco de tarifação para esse tipo de unidade possibilita um potencial de redução de consumo de energia, tendo em vista que o cliente passa a ser tarifado por três valores de energia divididos em três horários: fora-de-ponta, intermediário e ponta. Tais valores são estabelecidos por cada distribuidora dentro de limites estabelecidos pela ANEEL.

Para as unidades de média e alta tensão, foram revistos alguns conceitos de demanda, fator de potência e reativo, bem como tais parâmetros são tarifados dentro de uma fatura de energia. Modelos tarifários azul, verde e optante pelo grupo B foram esclarecidos.

Uma breve análise sobre a constituição de um sistema de monitoramento foi feita, tendo sido feita uma revisão sobre sensores, conversores analógico-digitais, *hardware* de processamento de dados e *software* de processamento e visualização dos dados.

3 SISTEMA PROPOSTO

Neste capítulo é apresentado o conceito envolvido no desenvolvimento do sistema de medição de unidades conectadas em média tensão, bem como a forma como os dados são coletados. *Hardware* utilizado, *firmware* e *software* desenvolvidos também são demonstrados neste capítulo. O protótipo desenvolvido foi instalado inicialmente na Fundação de Teleducação do Ceará (FUNTELC), localizado em Fortaleza/CE.

3.1 Conceito

Segundo a SEINFRA, o Estado do Ceará responde por 5441 unidades consumidoras, consumindo anualmente aproximadamente 550 MWh, totalizando um custo anual de aproximadamente 272 milhões de reais. Segundo publicação do MEC em Diário Oficial, o custo por aluno na educação básica no ano de 2018 foi de 2.091,37 reais por ano. Portanto, uma possível economia de 10% no valor gasto com contas de energia resultaria na possibilidade de financiamento de 13 mil novos estudantes no ensino básico.

O alto número de unidades consumidoras, bem como a ausência de sistemas de monitoramento ou tecnologias inteligentes de análise, ocasionam a existência de desperdícios no consumo de energia elétrica.

Outro motivo fomentador do desperdício se encontra na complexidade envolvida no faturamento de unidades de média tensão. Apesar de o número de unidades conectadas em média e alta tensão (1253 unidades) corresponder a 30% do número de unidades conectadas em baixa tensão (4188 unidades), a energia média consumida por unidades do grupo A (29.639 kWh/unidade) é aproximadamente 19 vezes o valor da média do consumo das unidades do grupo B (1.574 kWh/unidade). Portanto, a falta de conhecimento a respeito do modo de faturamento de uma unidade do grupo A pode impactar de forma significativa as contas do Estado.

Com base nesses dados, o desenvolvimento do sistema de monitoramento visa uma maior elucidação a respeito do que a unidade paga em uma conta de energia, proporcionando, possivelmente, uma redução do valor gasto.

3.1.1 Medidores eletrônicos de energia

O desenvolvimento de medidores eletrônicos de energia obedece a NBR 14520:2011, intitulada “Medidores eletrônicos de energia elétrica – Métodos de Ensaio” e a NBR 14519:2011, intitulada “Medidores eletrônicos de energia elétrica – Especificações”. Nela, constam termos, condições e metodologias de ensaios para a validação e confiabilidade do medidor.

De acordo com a NBR 14519, a classificação de um medidor eletrônico de energia deve obedecer às seguintes propriedades metrológicas: grandezas medidas (energia ativa, reativa, demanda), tecnologia de elemento(s) de sensor(es) de tensão e corrente, princípio de medição e tecnologia do mostrador (eletrônico/eletromecânico).

A faixa de operação dos medidores de energia eletrônicos varia entre 0,8 e 1,15 da tensão nominal, podendo operar em 50 Hz e 60 Hz.

Os medidores eletrônicos são classificados como A, B, C e D, sendo o critério dessas classificações sua tolerância ao erro. As Tabelas 3 e 4 indicam os erros de energia ativa e reativa, respectivamente, para medidores eletrônicos de energia separados por classe, em medidores monofásicos e polifásicos com cargas equilibradas.

Tabela 3 – Limites de erro percentual para medidores de energia ativa

		Limites percentuais para medidores com índice de classe			
%In	cos φ	D	C	B	A
10	1	± 0,2	± 0,5	± 1,0	± 2,0
100	1	± 0,2	± 0,5	± 1,0	± 2,0
100	0,5 Ind	± 0,3	± 0,6	± 1,0	± 2,0
100	0,8 Cap	± 0,3	± 0,6	± 1,0	± 2,0

Fonte: NBR 14520

Tabela 4 – Limite de erro percentual para medidores de energia reativa

		Limites percentuais para medidores com índice de classe			
%In	sen φ	D	C	B	A
10	1	$\pm 0,4$	$\pm 1,0$	$\pm 2,0$	$\pm 4,0$
100	1	$\pm 0,4$	$\pm 1,0$	$\pm 2,0$	$\pm 4,0$
100	0,5 Ind	$\pm 0,6$	$\pm 1,2$	$\pm 2,0$	$\pm 4,0$
100	0,8 Ind	$\pm 0,6$	$\pm 1,2$	$\pm 2,0$	$\pm 4,0$

Fonte: NBR 14520

Um dos medidores utilizados pela concessionária de distribuição do Estado do Ceará é o Landis+Gyr E750, que, seguindo a NBR 14522, disponibiliza uma saída de comunicação para leitura em tempo real de dados de consumo.

3.1.2 NBR 14522:2008

A NBR 14522:2008, intitulada “Intercambio de informações para sistemas de medição de energia elétrica” estabelece protocolos de comunicação inerentes ao desenvolvimento de um medidor eletrônico de energia. A referida norma discute a respeito de diversas saídas de comunicação presentes nos medidores. Entretanto, do ponto de vista deste trabalho, apenas a seção 3.4 da norma, que discute a respeito da saída de comunicação do usuário, é relevante.

O protocolo da saída do usuário é dividido em duas possibilidades: normal e estendida. No protocolo normal, o medidor envia um bloco de dados a cada segundo pela saída serial. A cada fim de intervalo de demanda, intervalos de duração de 15 minutos, o último bloco de dados desse intervalo é enviado três vezes consecutivas, sendo os dados repetidos uma vez por segundo.

A transmissão possui uma velocidade de transmissão de 110 baud com variação de 3%, assíncrona, monodirecional (apenas leitura), possui 1 *start bit* e 1 a 2 *stop bits*, 8 caracteres por bloco, 1 segundo de intervalo entre os blocos e possui nível lógico 1 com a saída desativada.

A saída serial fornece alguns dados relativos ao consumo de energia do cliente. Dentre eles, o tempo restante para o fim de cada intervalo, indicadores do modo atual de tarifação de energia reativa, se o horário atual é capacitivo ou indutivo, se o horário atual é ponta ou fora de ponta, tipo de tarifa, consumo de energia ativa e reativa, e um byte de CRC, para averiguação dos dados. O protocolo detalhado se encontra nos anexos deste trabalho.

Segundo a NBR 14522:2008, o protocolo estendido foi definido com o objetivo de fornecer todos os dados metrológicos e não apenas de faturamento. A informação é fornecida em 9 *bytes*, sendo utilizada a informação em quatro quadrantes para a obtenção de seu objetivo. Sendo assim, portanto, esse protocolo informa ao usuário dados de energia ativa direta (primeiro e quarto quadrante) e reversa (segundo e terceiro quadrante), e a indicação de qual quadrante pertence a energia reativa. O protocolo estendido difere-se também do protocolo normal pela verificação da veracidade de seus dados, sendo neste utilizado CRC16, enquanto que no protocolo normal utiliza-se CRC8.

Os pulsos de energia ativa e reativa resultantes do protocolo de medição são convertidos para valores em kWh e kVAr conforme a Eq. (4).

$$Energia (kWh/kVAr) = \frac{RTC \times RTP \times Ke}{1000} \times n^{\circ} \text{ de pulsos} \quad (4)$$

Onde:

- RTC é da relação de transformação do TC;
- RTP é a relação de transformação do TP;
- Ke é a constante estabelecida pelo fabricante do medidor;

Na FUNTELC, os valores de RTC, RTP e Ke são, respectivamente, 120, 2 e 0,3. Portanto, cada pulso representa o consumo de 72 Wh ou 72 VAr.

A partir dos valores de energia ativa e reativa, é possível inferir diversos cálculos de consumo do cliente, como demandas, fator de potência, UFER, DMCR, multas por ultrapassagem de demanda, entre outros.

Para este trabalho, apenas o protocolo de saída do usuário normal é utilizado.

3.2 Arquitetura de desenvolvimento

3.2.1 Hardware de aquisição dos dados

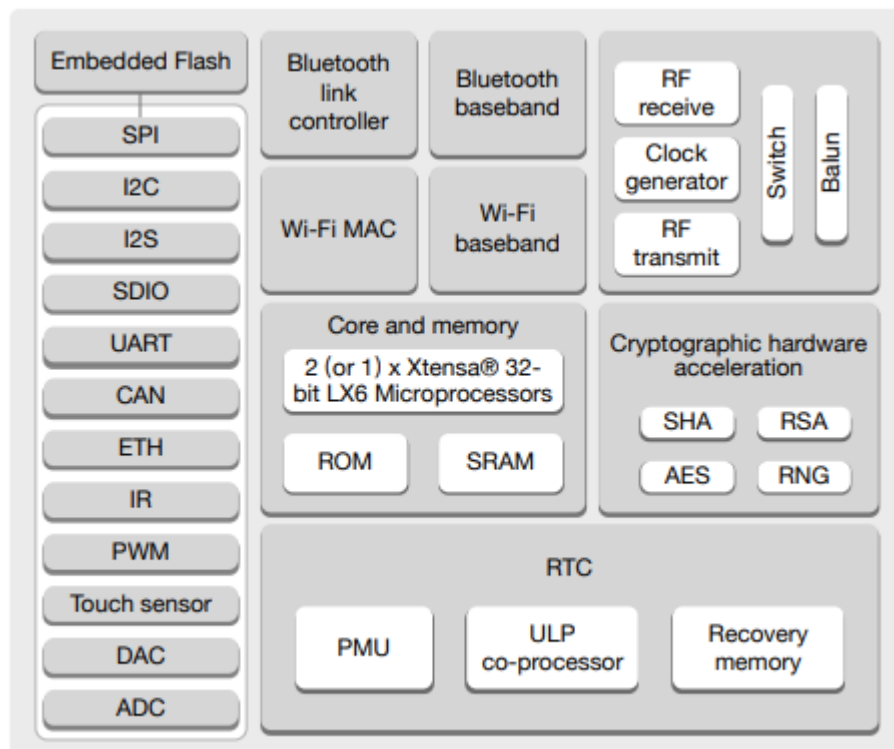
O *hardware* de desenvolvimento desenvolvido consiste basicamente na utilização de um microcontrolador ESP32. Seu baixo custo, baixa complexidade de uso e sua alta quantidade de funcionalidades fazem com que tenha crescido sua utilização em aplicações de celular, *wearable technologies* e de IoT.

O ESP32 é um módulo de baixo consumo, com frequência de 2,4 GHz, possuindo tecnologia *WiFi* e *Bluetooth*. Inclui um *switch* de antena, um *balun* de rádio frequência, amplificador de potência, filtros e módulo de gerenciamento de potência.

No modo WiFi, trabalha com os protocolos 802.11 b/g/n, 802.11 n (2.4 GHz), até 150 Mbps, WMM para envio de voz e vídeo, monitoramento de *beacon* para sistema de localização de alta precisão, 4 interfaces WiFi, entre outros. Pode ser *single-core* ou *dual-core*, com uma velocidade de processamento de até 600 MIPS. Possui também uma memória ROM de 448 KB, 520 KB de SRAM, oscilador interno de 8 MHz com calibração, oscilador externo variando entre 2 MHz e 60 MHz, dois grupos de timer, incluindo 2 timers de 64 bits e 1 watchdog timer.

Possui também 34 GPIOs programáveis, até 18 canais de conversores AD com resolução de 12 bits, 2 conversores DA com resolução de 8 bits, 10 sensores de toque, 4 SPI, 2 módulos I²S, 2 módulos I²C, 3 módulos UART, módulos PWM, sensor de efeito Hall, módulo de comunicação infravermelho, entre outras aplicações. A Figura 7 demonstra o diagrama de bloco funcional do ESP32.

Figura 7 – Diagrama de bloco funcional do ESP32



Fonte: ESP32 Series Datasheet, por Espressif

O ESP32 possui também o modo de programação *Over-the-air* (OTA), que possibilita que seu *firmware* possa ser atualizado à distância, sem a necessidade de uma conexão direta com o microcontrolador.

Biswas; Iqbal (2018) utilizam o ESP32 para monitorar um sistema de bombeamento de água utilizando energia fotovoltaica. Allafi; Iqbal (2017) utilizam o mesmo microcontrolador para o monitoramento de valores de tensão e corrente dos barramentos CC e CA de um sistema fotovoltaico. Sarjerao; Prakasarao (2018) utilizam o microcontrolador para o monitorar a poluição atmosférica em cidades da Índia. Rai; Rehman (2019) desenvolvem uma aplicação de transmissão de vídeo para sistemas de vigilância utilizando o dispositivo.

O ESP32 possui uma grande comunidade na Internet, o que facilita sua utilização. Além disso, é possível programá-lo na IDE do Arduino, possibilitando uma maior facilidade ao seu uso.

Neste trabalho, foi responsável, em resumo, pela leitura dados enviados pelo medidor, bem como seu envio para o *software*. Inicialmente, algumas configurações iniciais são feitas, como de GPIO, UART, Wi-Fi, OTA e *dual-core*. O *watchdog timer* é utilizado para a prevenção de erros.

O primeiro núcleo é responsável pela leitura dos dados, tratamento inicial e envio. Para cada bloco de dados recebido todo segundo, o microcontrolador faz uma leitura inicial dos dados e verifica se os dados estão corrompidos através do cálculo de CRC8. Caso estejam, envia um alerta ao usuário. De qualquer modo, os dados são enviados via Wi-Fi para o sistema de monitoramento. Para o envio dos dados, o ESP32 utiliza requisições GET. O GET é um verbo do protocolo HTTP utilizado para solicitar dados de uma fonte específica. Através da requisição GET, é possível realizar o envio de dados como parâmetros. Os parâmetros enviados constam após a URL da fonte, como consta na Figura 8.

Figura 8 – Exemplo de requisição GET

```
www.meusite.com.br/index.html?name1=value1&name2=value2
```

Fonte: Próprio autor

Conforme pode ser visto na Figura 8, o início dos parâmetros é marcado pela presença do caractere “?”, e a adição de novos parâmetros é feita utilizando o caractere “&”. No exemplo mostrado na Figura 8, o parâmetro “name1” tem o valor de “value1”, enquanto que o parâmetro “name2” tem o valor de “value2”.

Os dados são enviados da forma mostrada na Figura 8, contendo como parâmetros os 8 *bytes* enviados pelo medidor e o valor do CRC8 calculado pelo microcontrolador.

Um sistema de *backup* foi desenvolvido no *firmware* do microcontrolador, para quando estiver impossibilitado de enviar os dados por algum motivo, que consiga enviá-los assim que a operação estiver normalizada.

O *hardware* também analisa outras possíveis falhas, como falta de conexão com a rede, ausência de dados recebidos ou impossibilidade de envio dos dados. Caso haja a detecção de algum desses erros, um alerta é enviado por e-mail ao usuário.

Os dados são enviados a cada 2 minutos para o servidor, e quando o período de demanda se encerra. A biblioteca WiFiManager foi utilizada para a configuração do ESP32 como *Access Point*, cadastro de SSID e senha da rede Wi-Fi.

O segundo núcleo do ESP32 analisa a cada 10 segundos se o usuário solicitou *reset* do microcontrolador. Tal aplicação visa a possível necessidade de reiniciar o dispositivo em caso de alguma falha.

A distribuidora de energia local exige que a comunicação entre o microcontrolador e o medidor de energia seja feita utilizando um acoplador óptico, de modo a isolar eletricamente o medidor de qualquer dispositivo que venha a ele ser conectado.

Por obter os dados diretamente do medidor de energia, o sistema não necessita de sensores, o que reduz o seu custo de desenvolvimento. Também por esse fato, os dados possuem uma precisão de 100%, tendo em vista que os mesmos dados utilizados para o faturamento são usados no sistema de medição.

O código implementado no ESP32 encontra-se nos apêndices deste trabalho.

3.2.2 Software de desenvolvimento

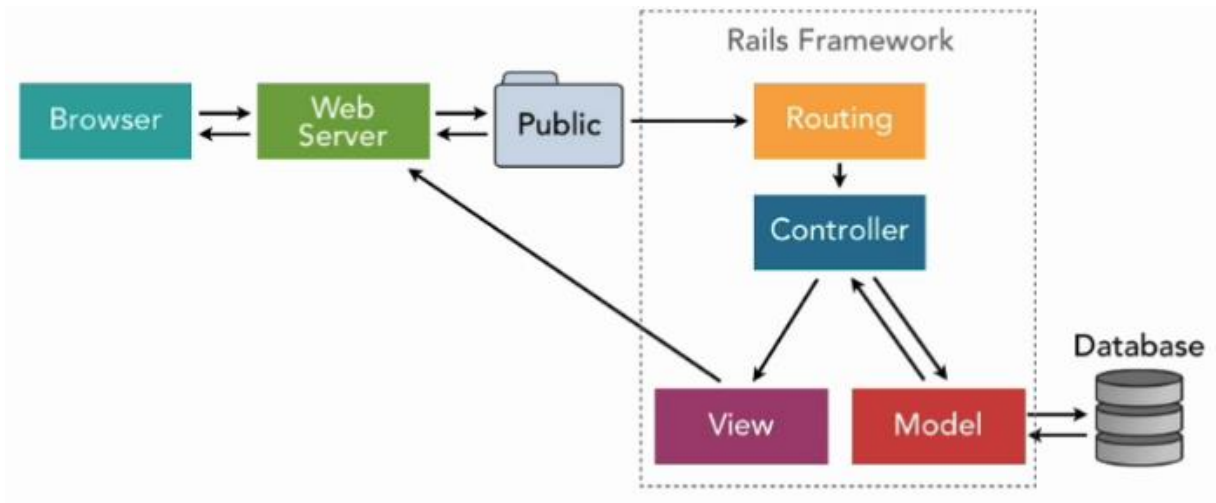
O *software* do sistema de monitoramento foi desenvolvido utilizando a linguagem de programação orientada a objeto *Ruby*, com auxílio do *framework Ruby on Rails* (RoR).

RoR é um *software* livre, com uma grande comunidade ativa que contribui para o seu desenvolvimento. A última versão lançada foi a 5.2.3. O desenvolvimento de sua sexta versão encontra-se na fase beta.

O RoR baseia-se na ideia de “Convenção sobre Configuração”. Com isso, convenções criadas pelo *framework* reduzem a necessidade de tempo gasto com configurações no desenvolvimento, liberando mais tempo para o desenvolvimento real da aplicação.

O desenvolvimento em RoR é baseado na arquitetura Modelo-Visão-Controlador (MVC). Os modelos são os dados da aplicação, onde consta toda a parte lógica, regras e funções dos dados do sistema. Os controladores mediam as requisições ao sistema, transformando-as em comandos para as visões e modelos. A visão mostra apenas as informações úteis ao usuário da aplicação. A Figura 9 detalha visualmente o funcionamento da arquitetura MVC do RoR.

Figura 9 – Arquitetura MVC do RoR



Fonte: Ruby on Rails, HTTP, MVC and Routes, <www.medium.com>

O *software* desenvolvido em *Ruby* para este trabalho possui diversas finalidades. A princípio, os dados são recebidos pela requisição GET enviada pelo ESP32. Ao receber os dados, um tratamento é feito, de modo a obter especificidades de cada dado recebido, como tempo restante até o fim do intervalo de demanda, se o período atual exige controle de reativo indutivo ou capacitivo, se é horário de ponta ou fora-de-ponta, qual o modo de tarifação que o cliente é tarifado e valores de energia ativa e reativa.

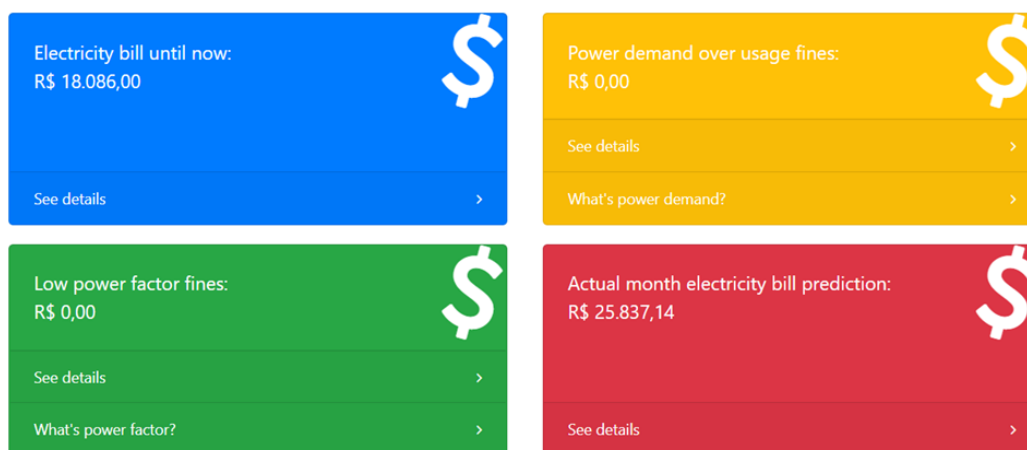
O gerenciador de banco de dados utilizado foi o PostgreSQL, desenvolvido também em código aberto. A aplicação desenvolvida foi hospedada em uma VPS, que é um servidor responsável capaz de processar o código desenvolvido, com espaço de disco, memória RAM e tráfego completamente dedicados.

A partir desses dados, toda a inteligência do sistema é processada. Inicialmente a validade dos dados é verificada através do CRC8, sendo salvos somente em caso afirmativo. Em caso negativo, um e-mail de alerta é enviado ao usuário. O componente ActionMailer do RoR é utilizado para o envio dos e-mails.

Análise de previsão de consumo é feita, através de uma regra de três simples considerando os valores de consumo até o dia do mês atual. O sistema também realiza uma análise de previsão de ultrapassagem de demanda ativa, considerando o valor de atual e o tempo restante para o fim do intervalo de demanda. Considerando o consumo mensal atual, o sistema analisa também se houve alguma ultrapassagem de demanda ou pagamento de multa por consumo excedente de energia reativa, citando também os devidos valores. A Figura 10 demonstra os valores acima citados em 16 de março de 2019. O desenvolvimento da Figura 10

foi realizado com auxílio da biblioteca *Bootstrap*, uma biblioteca CSS, utilizado para personalização e desenvolvimento *front-end*.

Figura 10 – Dados iniciais extraídos do sistema



Fonte: Próprio autor

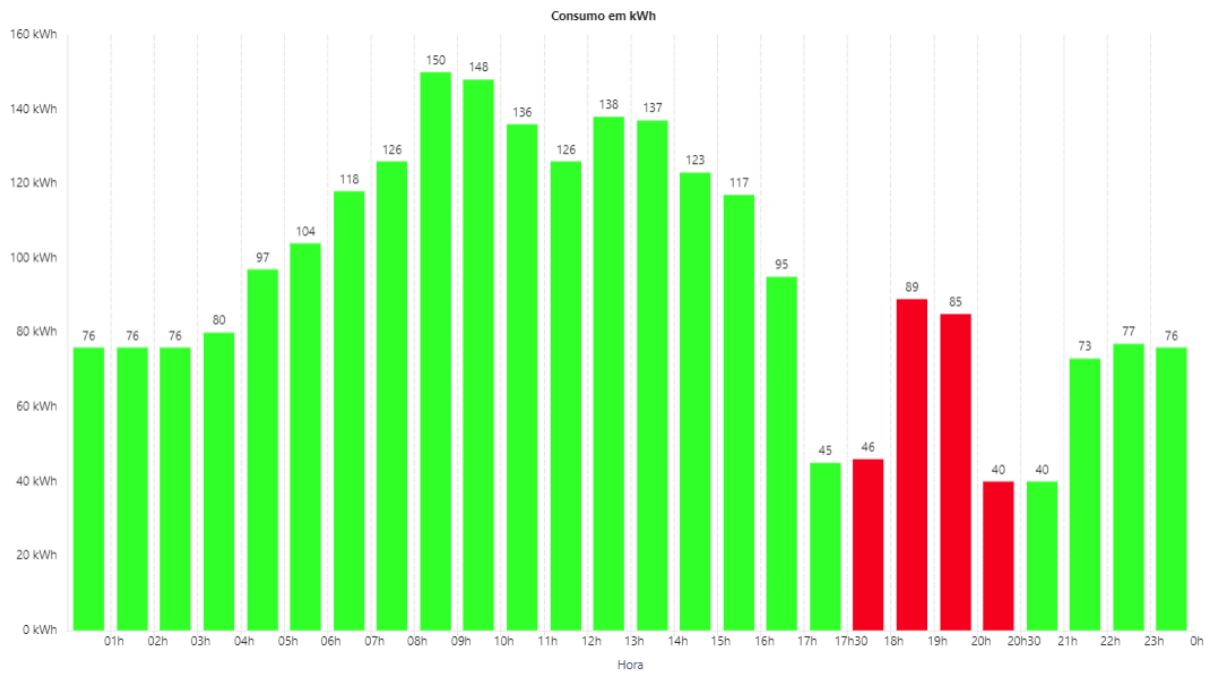
A aplicação desenvolvida trabalha em tempo real. Portanto, a cada 2 minutos, os dados e gráficos da aplicação são atualizados, sem a necessidade de recarregar a página. Para isso, foi utilizada trabalha com auxílio do protocolo WebSocket, o qual mantém uma conexão com o usuário aberta sem a necessidade de uma requisição. Para que o protocolo WebSocket fosse compatível com o RoR, foi desenvolvido o Action Cable, que fornece um *framework* no lado do cliente em JavaScript, e no lado do servidor, em *Ruby*.

No sistema desenvolvido, era possível também observar gráficos em tempo real de diversas variáveis, com consumo ativo, reativo, fator de potência e demanda, detalhados por dia, mês e ano. A Figura 11 mostra o consumo de energia ativa em 12 de maio de 2019. O sistema possui um gráfico similar ao apresentado na Figura 11, sendo que com valores de energia convertidos em reais, levando em consideração a tarifa, o horário e o consumo da unidade. Os dados em vermelho representam o consumo no horário de ponta.

A Figura 12 detalha o consumo de energia ativa separado por dia durante o mês de abril de 2019, e a Figura 13 mostra o consumo por mês no ano de 2019.

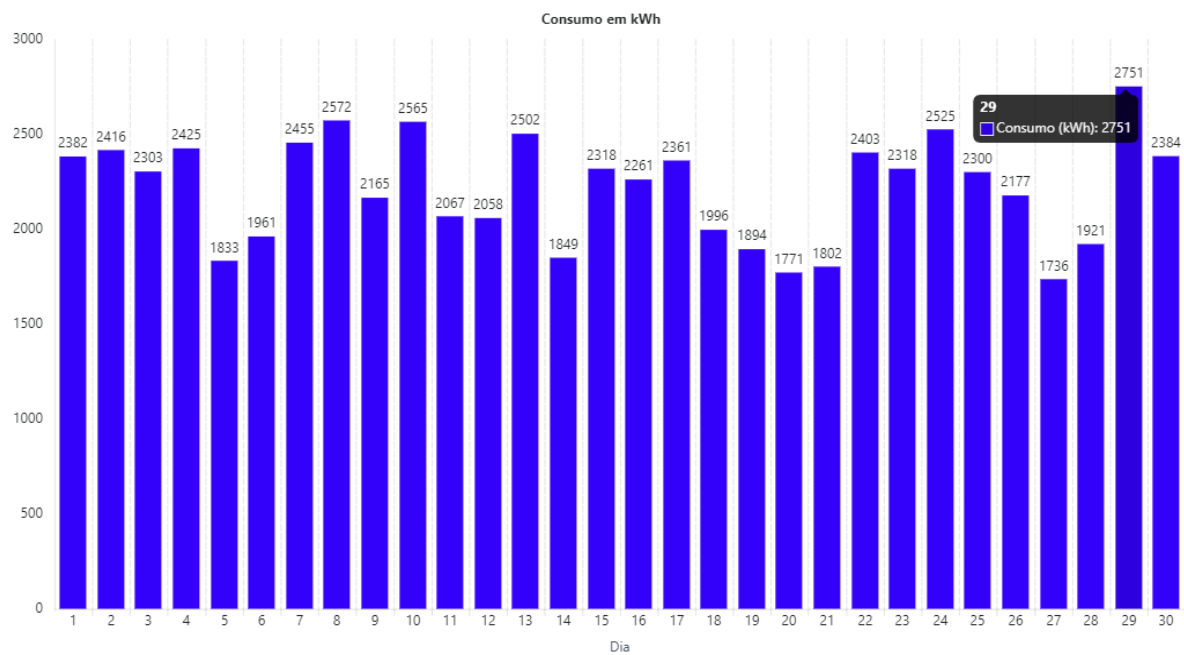
Na Figura 14 consta o gráfico da demanda ativa no dia 3 de maio de 2019. Para o cálculo da demanda, o sistema multiplica o consumo em cada intervalo de 15 minutos por 4, tendo em vista que a demanda representa a potência média nesse intervalo. 96 valores de demanda por dia são representados em cada gráfico.

Figura 11– Consumo de energia ativa em 12 de maio de 2019



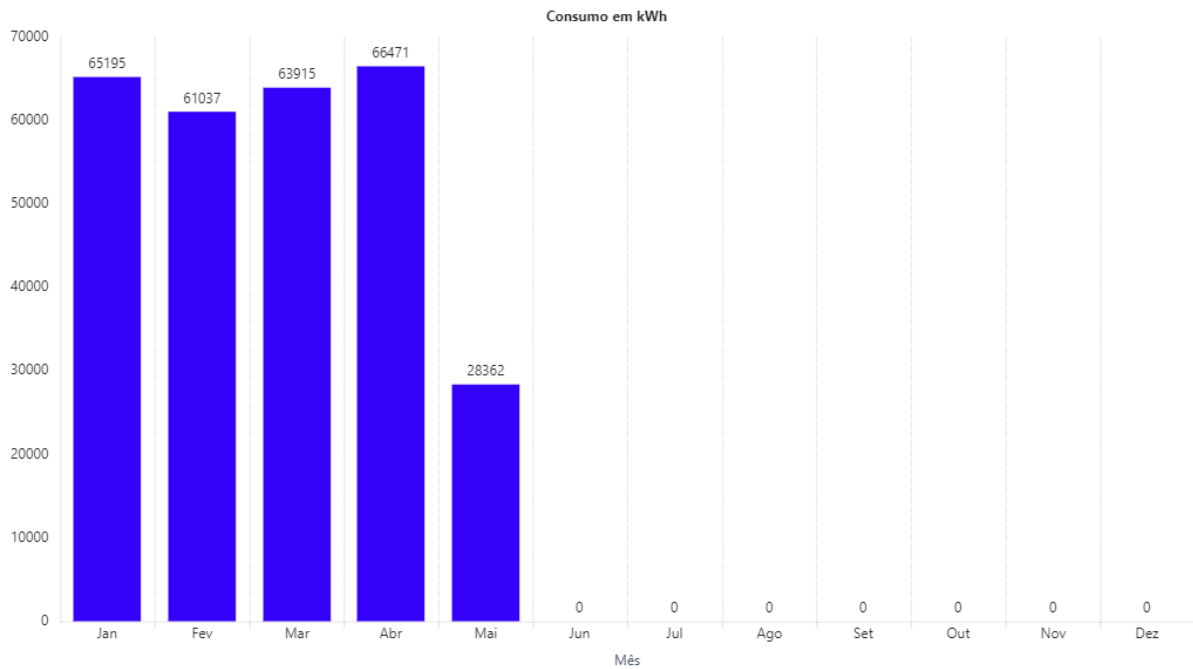
Fonte: Próprio autor

Figura 12 – Consumo de energia ativa no mês de abril de 2019



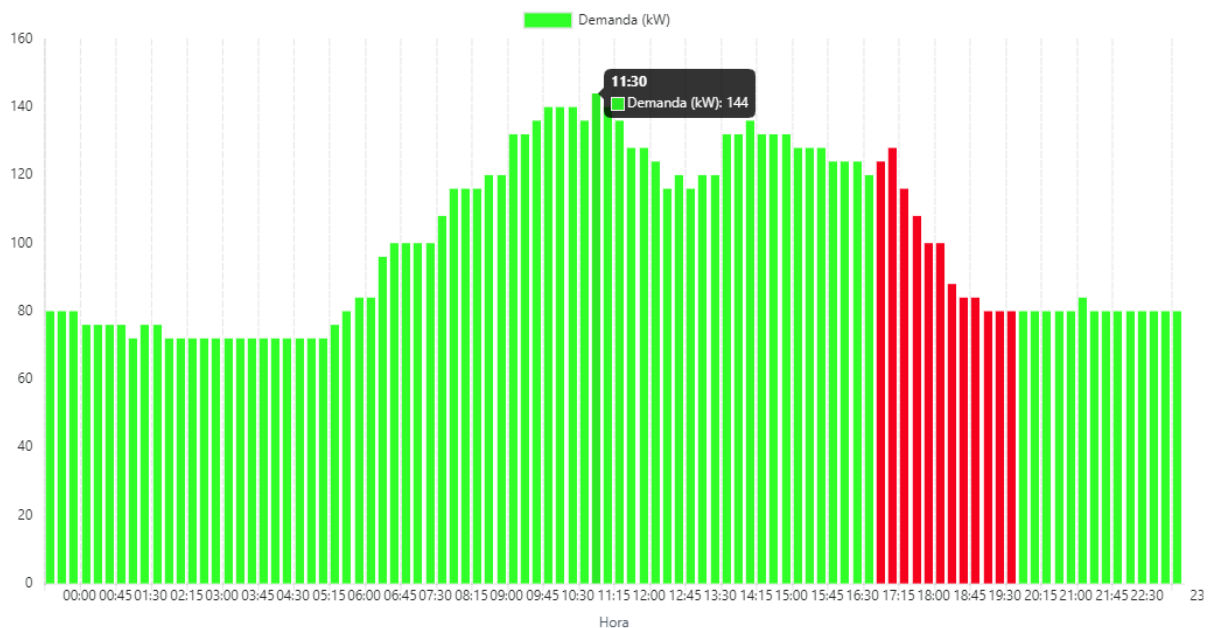
Fonte: Próprio autor

Figura 13– Consumo de energia ativa em 2019



Fonte: Próprio autor

Figura 14– Demanda ativa em 3 de maio de 2019

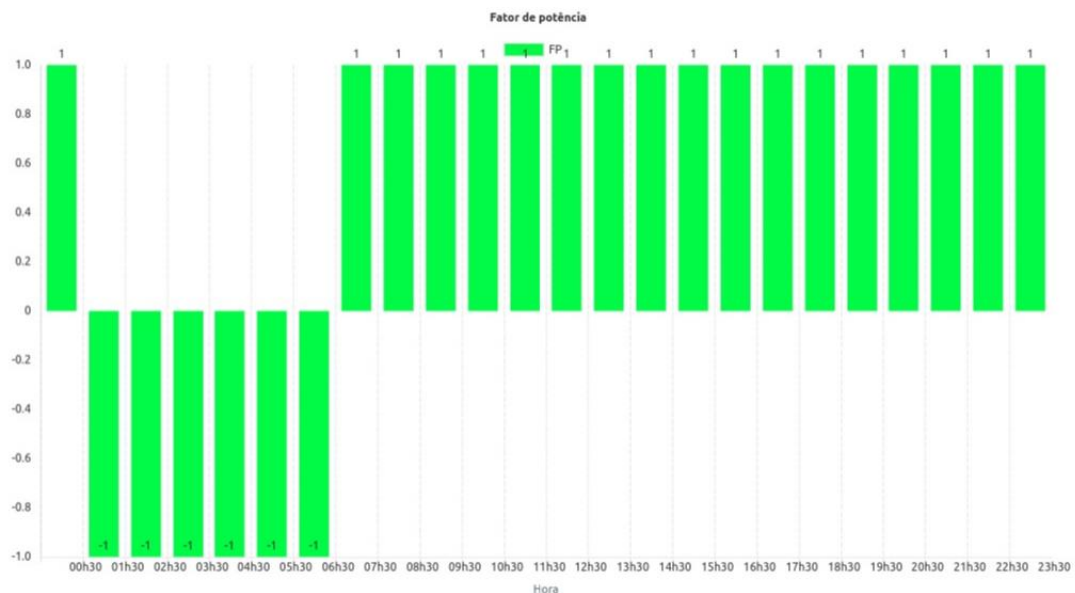


Fonte: Próprio autor

O monitoramento de energia reativa possui uma ressalva. No momento em que o dispositivo desenvolvido foi conectado ao sistema de medição, verificou-se que a energia reativa disponibilizada através do protocolo de comunicação refere-se somente ao tipo analisado no referido período, i.e., no período em que o fator de potência essencialmente

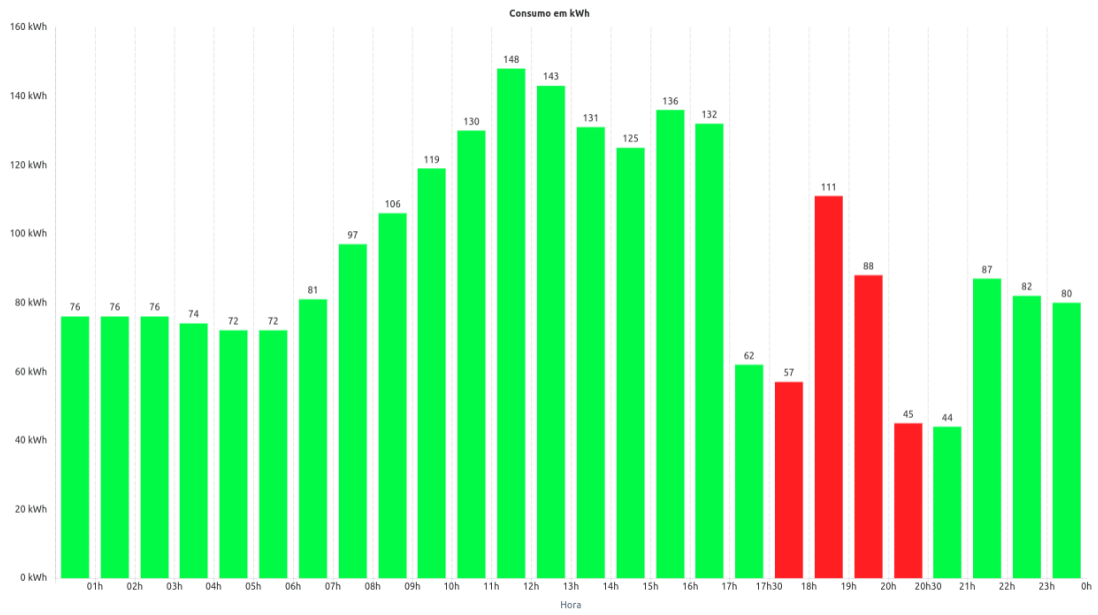
capacitivo é passível de punição, normalmente durante a madrugada, apenas dados referentes à energia reativa capacitiva são fornecidos, enquanto que nos demais períodos, apenas a energia reativa indutiva é analisada. Portanto, do ponto de vista de tarifação, o sistema é eficaz ao detectar algum consumo excedente de energia reativa e multa consequente. Para o cálculo do fator de potência, foram utilizados valores de energia ativa e reativa em cada hora. Para a diferenciação do fator de potência considerado em horários indutivos e capacitivos, neste foi adicionado o sinal negativo ao cálculo. A Figura 15 demonstra o fator de potência horário no dia 10 de maio de 2019. Para efeito comparativo, as Figuras 16 e 17 demonstram os valores de consumo de energia ativa e reativa no mesmo dia, respectivamente.

Figura 15– Fator de potência no dia 10 de maio de 2019



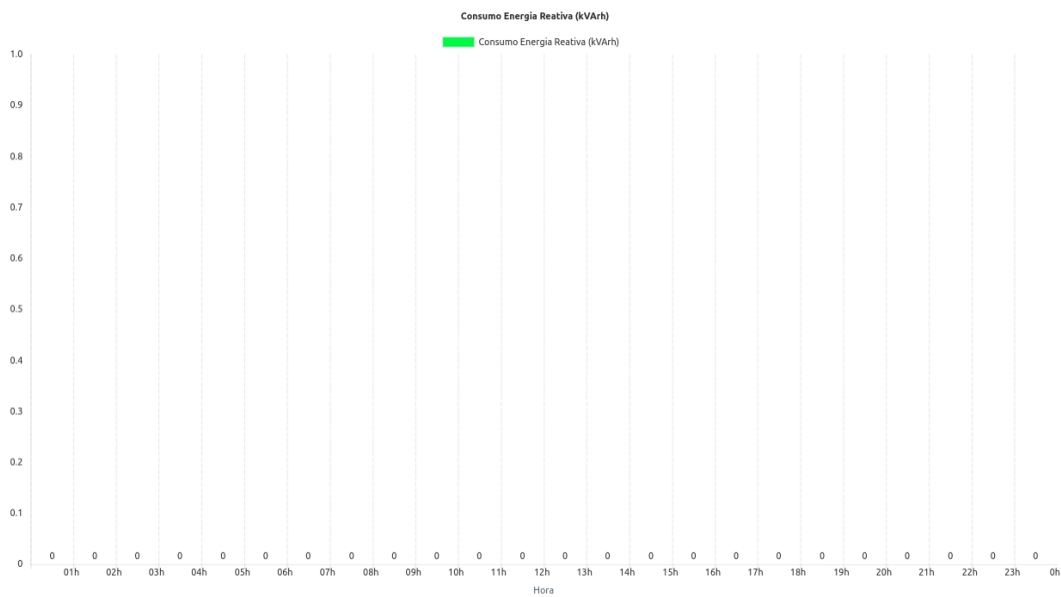
Fonte: Próprio autor

Figura 16 - Consumo de energia ativa no dia 10 de maio de 2019



Fonte: Próprio autor

Figura 17 - Consumo de energia reativa no dia 10 de maio de 2019



Fonte: Próprio autor

No sistema não foi identificado consumo considerável de energia reativa, o que explica a constante proximidade do fator de potência ao valor unitário.

Foi desenvolvido também uma página onde possíveis melhorias são analisadas. Nesta página é solicitado inicialmente durante qual período o consumo deverá ser analisado. Em seguida, diversas análises são feitas, essencialmente do ponto de vista da demanda, da tarifa, do consumo de energia reativa e do consumo de energia ativa.

Do ponto de vista da demanda, é verificado se houve ultrapassagem de demanda no período estabelecido. Em caso afirmativo, o sistema detalha todos os momentos em que houve ultrapassagem de demanda, citando valor de demanda registrado em kW, hora e dia e valor da multa a ser pago. O fator de carga (FC), que é a razão entre a demanda média e demanda máxima em um determinado período, também é analisado, verificando se tal valor está otimizado. Quanto mais próximo de 1, mais otimizado é o consumo. Entretanto, o sistema considera o valor de 0,6 como um valor ótimo. O sistema verifica também se há uma contratação de demanda acima do necessário, citando o potencial de redução em kW e o potencial de economia em reais em caso afirmativo.

Considerando o modo de tarifação, o sistema analisa, com base nos três modelos tarifários possíveis para uma unidade do grupo A, qual a melhor tarifação para o usuário.

Sobre o consumo de energia reativa, o sistema analisa se houve algum consumo excedente de energia reativa, citando o valor da multa a ser pago. Além disso, verifica também se houve cobrança de multa por ultrapassagem de demanda reativa, citando também, em caso afirmativo, os horários em que ocorreu e o valor da multa a ser pago.

Com relação ao consumo de energia ativa, o sistema verifica inicialmente o percentual de energia ativa consumida no período da madrugada com relação ao consumo total. Um alto consumo no período da madrugada pode significar cargas não essenciais ligadas nesse período. Além disso, o sistema faz uma análise considerando o consumo de energia no período de ponta e fora de ponta, citando o percentual do valor pago em energia em cada horário, bem como os respectivos valores em reais.

O sistema desenvolvido também possui um sistema de alertas, enviados ao administrador em caso de alguma anormalidade detectada. Considerando anormalidades no funcionamento do *hardware*, o sistema é capaz de detectar falta de envio de dados e corrupção de dados, falta de comunicação entre medidor e *hardware*. Do ponto de vista do consumo, o sistema é capaz de prever, através de uma regra de três considerando o consumo atual e o tempo para o fim do intervalo de demanda, se haverá alguma ultrapassagem de demanda, enviando

um e-mail antes que a ultrapassagem se concretize. O sistema também é capaz de alertar quando detectar o consumo excedente de energia reativa.

Além disso, um e-mail contendo um relatório é enviado ao usuário ao fim de cada dia, informando dados referentes ao consumo de energia no dia anterior, como consumo total, consumo no horário de ponta e fora de ponta em reais e kWh, valor máximo de demanda registrado, se houve alguma multa por ultrapassagem de demanda ou reativo, entre outros diversos fatores. Um relatório similar é enviado ao usuário ao fim de cada mês, constituído essencialmente pelas mesmas informações, mas contendo também uma análise tarifária, sobre quanto teria sido gasto em cada modelo tarifário e qual teria sido a melhor opção considerando aquele mês.

3.3 Resultados obtidos

Com a instalação do sistema desenvolvido no FUNTELC, alguns resultados práticos foram obtidos.

Inicialmente, o sistema detectou uma utilização excessiva de energia ativa durante a madrugada no período analisado, aproximadamente 30% do total, o que poderia ser resultante de desperdício. Entretanto, foi identificado que o consumo excessivo do neste período origina-se da antena de transmissão do sinal televisivo, que permanece ligada ininterruptamente.

Além disso, no mês de dezembro de 2018, a FUNTELC foi multada por um consumo excessivo de energia reativa. Após análise no sistema, tal consumo não foi identificado. Assim, o sistema contribuiu para o questionamento e posterior confirmação do equívoco.

O sistema identificou também que não há significativas variações na conta de energia com uma possível mudança do modo de tarifação azul para o verde.

Foi identificado também que a unidade consumidora contrata um valor de demanda acima do necessário. Por ser tarifada no modo azul, a unidade contrata dois valores de demanda, um para o horário de ponta e outro para o horário fora de ponta. Os dois valores estavam contratados acima do necessário, conforme Figura 18.

Figura 18– Análise de demanda contratada no período de dezembro de 2018

	Demanda contratada (kW)	Demanda registrada (kW)	Data e hora	Potencial de redução	Potencial de economia
Fora Ponta	205	144	13/12/2018 às 16:15	61 kW	R\$ 848,51
Ponta	180	132	13/12/2018 às 17:45	48 kW	R\$ 1.111,20

Fonte: Próprio autor

Observando a Figura 18, é possível observar que um potencial de redução de demanda contratada de 61 kW para o horário fora de ponta e de 48 kW para o horário de ponta puderam ser obtidos, resultando num potencial de economia de aproximadamente 10% do valor total da conta de energia.

3.4 Conclusão parcial

Neste capítulo, o conceito em que o sistema de monitoramento se baseia foi apresentado. Foi feita uma breve revisão a respeito dos medidores eletrônicos de energia e das normas que regem seu desenvolvimento e protocolos de comunicação. O protocolo que consta na NBR 14519:2008, essencial para o desenvolvimento deste trabalho, foi revisado, e todas informações possíveis de serem extraídas do medidor foram apresentadas. A parte relevante da norma se encontra nos anexos deste trabalho.

Foi apresentado a arquitetura do sistema, iniciando pelo microcontrolador utilizado, suas características físicas e modos de operação. Sua funcionalidade na aplicação deste sistema também foi analisada. Uma breve revisão a respeito das funcionalidades implementadas no seu código também foi feita.

Em seguida, o *software* desenvolvido em Ruby foi apresentado. Tal software é responsável por toda a inteligência do sistema, desde o recebimento dos dados, tratamentos e visualização por parte do usuário. As informações essenciais para o usuário foram apresentadas. O sistema de alerta de e-mails foi explanado.

4 POSSÍVEIS APLICAÇÕES DO SISTEMA DE MONITORAMENTO

Neste capítulo, algumas abordagens cabíveis a sistemas de monitoramento de energia serão abordadas.

4.1 Modelagem de perfil de consumo

A partir da obtenção de dados de consumo, é possível inferir, através de modelos de análise de processo estocásticos, um perfil de consumo. Tendo em vista que o usuário do sistema de monitoramento também pode ser a distribuidora de energia, tais inferências podem ser úteis para o planejamento de expansão da rede e para estimar o consumo durante um período de falha do medidor ou de furto de energia Groß (2017), auxiliando a distribuidora em uma eventual cobrança em caso de falhas. Além disso, considerando que o usuário final seja o próprio consumidor de energia, processos estocásticos podem auxiliar na previsão do consumo de energia. A mudança recente no perfil de consumo com o advento de novos tipos de carga, como os carros elétricos, e o aumento exponencial da geração de energia na rede de distribuição também motivam a necessidade do estudo de modelagem de perfil de consumo. Um dos modelos estocásticos mais analisados para esse tipo de problema é o método das cadeias de Markov.

4.1.1 Cadeias de Markov

Cadeias de Markov estudam a probabilidade de um evento acontecer, considerando que o próximo evento possui dependência apenas do evento atual, sendo independente do histórico restante do processo.

Uma cadeia de Markov finita tem N estados possíveis, sendo N um número inteiro finito. O conjunto de todos os estados possíveis é representado vetor $E = \{1, 2, \dots, N\}$. O espaço de tempo em que uma amostra é realizada é chamado de *evento*, representado por n , e o espaço de tempo entre os eventos é chamado de *período*.

A variável aleatória X_n representa o estado no evento n . A sequência aleatória de uma cadeia de Markov é representada por $\{X_0, X_1, X_2, \dots\}$. O índice subscrito indica, portanto, em qual evento o respectivo estado ocorreu.

A Figura 19 retrata a representação de uma cadeia de Markov.

Figura 19– Representação de uma cadeia de Markov

X_0	X_1	X_2	...	X_n	...	Estado
Período 1	Período 2	... Período n	
0	1	2	...	n	...	Evento

Fonte: Próprio autor

Se em um evento n , a cadeia estiver no estado i , então afirma-se que $X_n = i$. Portanto, a probabilidade de que durante o evento n a cadeia esteja no estado i é descrita como $P(X_n = i)$. A probabilidade de a cadeia estar no estado j no próximo evento pode ser descrita como $P(X_{n+1} = j)$. A condição probabilística de que o sistema esteja no estado j no evento $n+1$, considerando que esteja no estado i no evento n , pode ser descrito, portanto, conforme a Eq. (5).

$$p_{ij} = P(X_{n+1} = j | X_n = i) \quad (5)$$

A variável p_{ij} representa, portanto, uma probabilidade de transição entre o estado i e o estado j . Considerando a Eq. (5), e considerando o fato de que, por definição, em uma cadeia de Markov o estado do próximo evento depende apenas do estado atual, é possível afirmar a Eq. (6).

$$P_{ij} = P(X_{n+1} = j | X_n = i) = P(X_{n+1} = j | X_1 = a, X_2 = b, \dots, X_n = i) \quad (6)$$

Os valores de $X_0, X_1, X_2, \dots, X_{n-1}$ podem, portanto, ser ignorados.

Outra propriedade da cadeia de Markov está no fato de as transições serem invariáveis com o tempo. Portanto, é possível afirmar a Eq. (7).

$$p_{ij} = P(X_{n+1} = j | X_n = i) = P(X_1 = j | X_0 = i) \quad (7)$$

A matriz de transição de uma cadeia de Markov de N estados possui dimensão de $N \times N$, normalmente denominada por P . A Figura 20 demonstra um exemplo de uma matriz de Markov com 4 estados possíveis.

Figura 20– Exemplo de uma matriz de transição de uma cadeia de Markov

$P =$	$X_n \backslash X_{n+1}$	1	2	3	4
	1	p_{11}	p_{12}	p_{13}	p_{14}
	2	p_{21}	p_{22}	p_{23}	p_{24}
	3	p_{31}	p_{32}	p_{33}	p_{34}
	4	p_{41}	p_{42}	p_{43}	p_{44}

Fonte: Sheskin (2011)

Com base na Figura 20, as linhas, representadas pelo primeiro número do índice subscrito, consideram o estado atual, enquanto que as colunas, representadas pelo segundo número subscrito, representam o próximo estado. As probabilidades representadas em cada linha totalizam 100%, considerando que a transição deve ocorrer entre alguma das possibilidades existentes.

Cadeias de Markov podem ser utilizadas para a modelagem de perfil de carga elétrica. Labeeuw; Deconick (2013) apresentam um modelo que utiliza cadeias de Markov para criar perfis de consumo de energia elétrica em unidades residenciais, e outras cadeias de Markov para simular perfis aleatórios de consumo. Uhrig (2014) compara métodos com e sem cadeias de Markov em processo de modelagem de consumo de energia elétrica. Groß (2017) utiliza cadeias de Markov e regressão linear para gerar perfis de consumo de energia elétrica em unidades residenciais, concluindo que o processo de modelagem por Markov se assemelha com maior precisão aos dados originais.

4.1.1.1 Modelagem

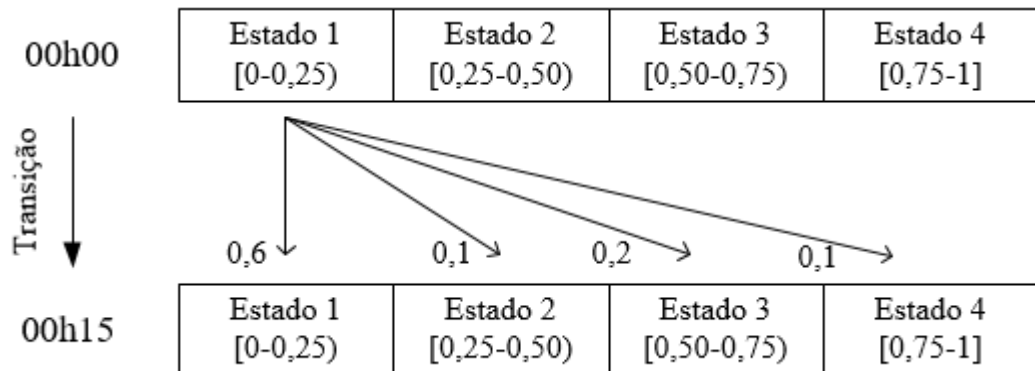
O início de operação do sistema de monitoramento deu-se em 02 de dezembro de 2018. Como os dados resultantes representam o consumo de energia em intervalos de 15 minutos, 96 valores de consumo são obtidos por dia. A modelagem representada neste trabalho foi obtida a partir de valores de consumo de energia obtidos em dias úteis, para que a representação fosse mais homogênea e a análise qualitativa do método desenvolvido fosse mais coerente. Considerando tais premissas, aproximadamente 10 mil dados foram utilizados para a modelagem do sistema, que foi realizada com o *software* MATLAB.

Inicialmente foram definidos os possíveis estados do modelo. Para isso, os valores de energia em kWh foram normalizados, gerando valores entre 0 e 1. A quantidade de estados é livre, entretanto, quanto maior, mais realista a simulação se torna.

Por exemplo, de forma simples, considerando que a quantidade de estados é 4, os possíveis estados em um conjunto de dados normalizados encontram-se entre $[0,0-0,25)$, $[0,25-0,50)$, $[0,50-0,75)$ e $[0,75-1]$. O consumo, portanto, em cada um dos 96 intervalos de tempo dentro de um dia, pode ser estabelecido dentro de um desses 4 intervalos. Os 96 intervalos diários (00h00 a 00h15, 00h15 a 00h30..., 23h45 a 00h00) foram indexados, com valores variando de 1 a 96.

A matriz de transição analisa a probabilidade de em determinado horário (00h00 por exemplo), ao averiguar um consumo com estado i , que o consumo no próximo horário, 00h15 no mesmo exemplo, seja j . Para determinar a matriz de transição deverá ser analisado, portanto, o estado do consumo de todos os horários, bem como o consumo do horário seguinte. A Figura 21 representa um exemplo de uma representação do processo de modelagem.

Figura 21– Representação do processo de modelagem



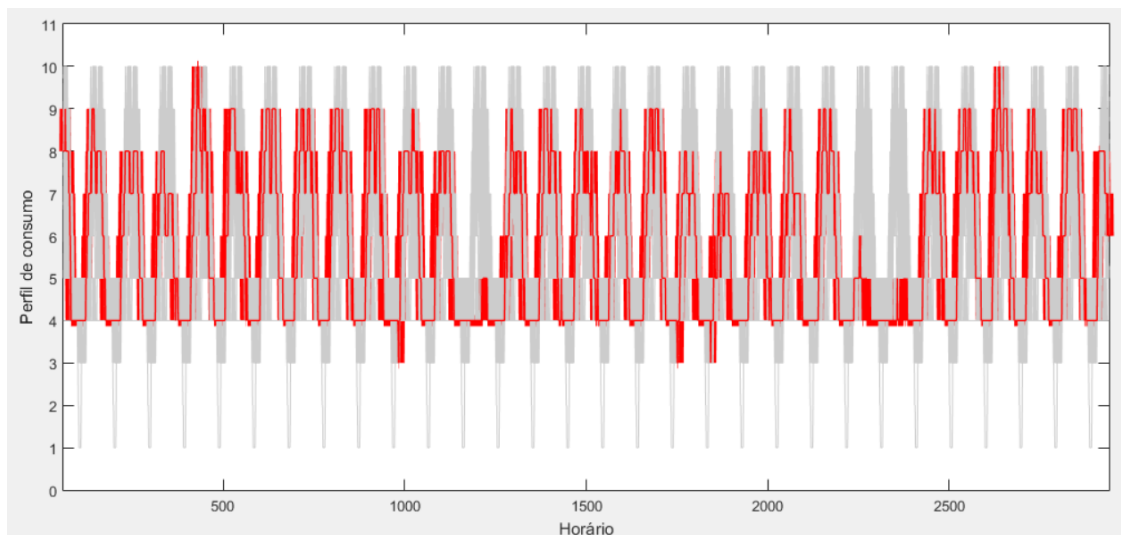
Fonte: Próprio autor

A partir da Figura 21, a probabilidade de, durante o evento 1, o estado continuar no estado 1 é de 60%, enquanto que para mudar para os estados 2 e 4 tem probabilidade de 10% e 20% de migrar para o estado 3. Deve-se atentar ao fato que, cada horário possui uma matriz de transição de estado, e por haver 96 transições diárias, a matriz de transição, para este exemplo, terá dimensão $4 \times 4 \times 96$.

Na utilização dos dados obtidos com o sistema de monitoramento, foi utilizado um conjunto de estados com 10 soluções possíveis ($\{[0-0,1), [0,1-0,2), \dots, [0,9-1]\}$), resultando em uma matriz de transição com dimensão $10 \times 10 \times 96$. Portanto, foi identificado para cada evento n seu estado e qual o estado no evento $n+1$, obtendo assim a matriz de probabilidade com base em todas os eventos ocorridos.

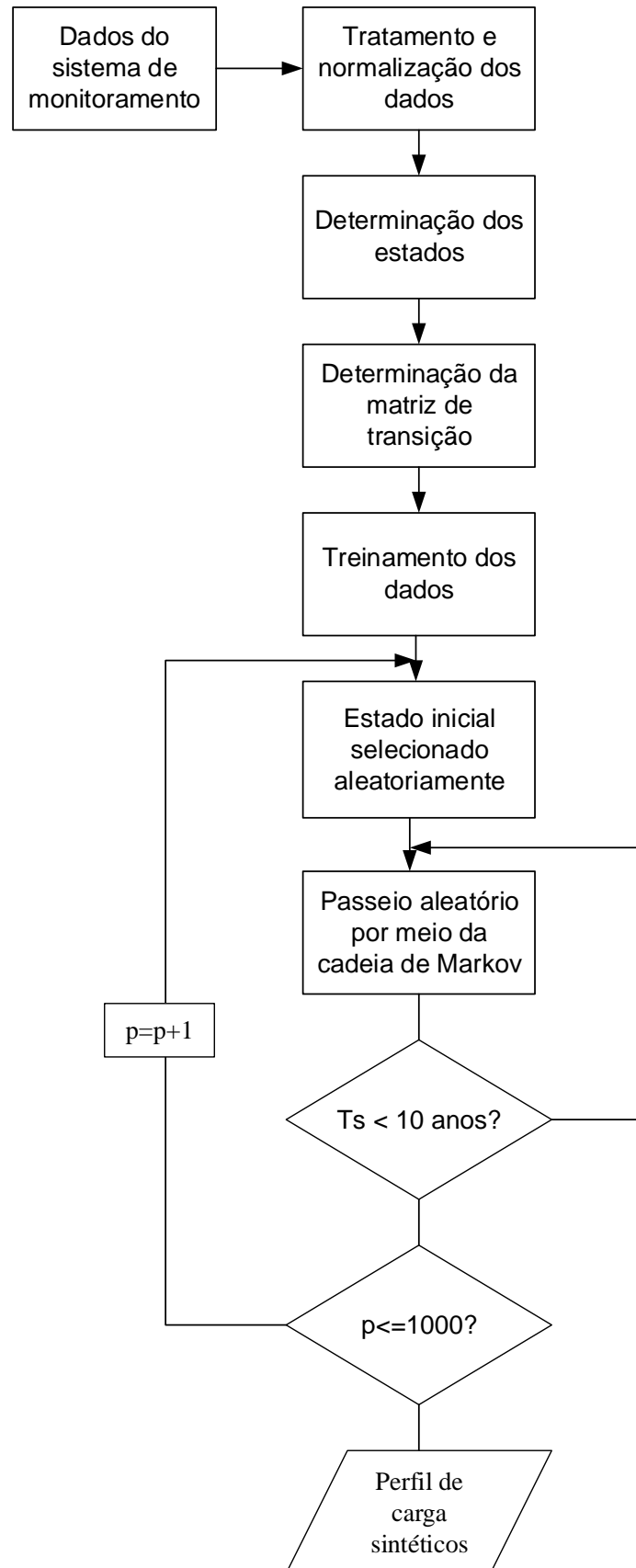
Após a obtenção da matriz de transição, dá-se início ao processo de simulação. É simulado 1000 vezes com auxílio de um passeio aleatório a matriz de transição encontrada, durante um período equivalente de 10 anos. Com os dados encontrados, apenas os últimos 6 meses são considerados em análise, tendo em vista que o perfil traçado através da matriz de probabilidades tende a se aproximar de valores característicos com o avanço do período de simulação. A partir dos perfis encontrados, análises de verificação de exatidão da matriz de probabilidade definida deve ser analisada. O código desenvolvido no MATLAB encontra-se nos apêndices deste trabalho. A Figura 22 representa o modelo extraído após a simulação realizada com o MATLAB, e a Figura 23 representa o fluxograma do modelo desenvolvido.

Figura 22 – Perfil de consumo obtido através de simulação com MATLAB



Fonte: Próprio autor

Figura 23 – Fluxograma da modelagem desenvolvida

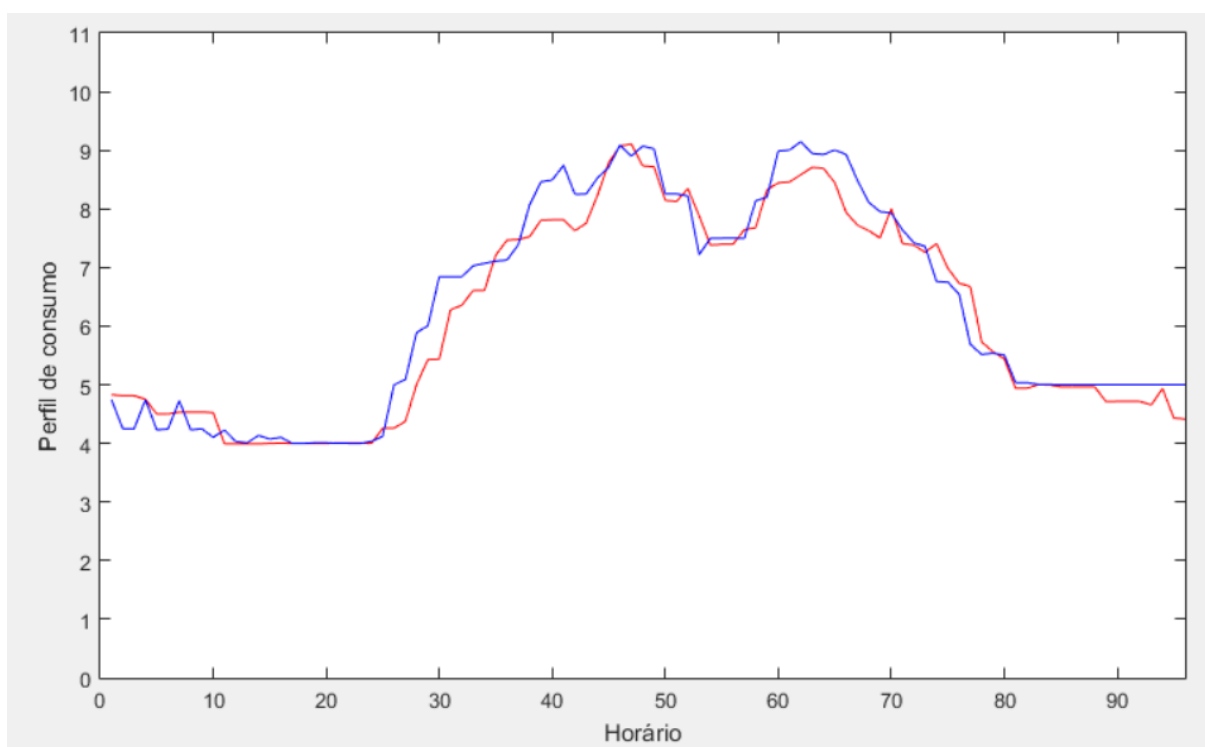


Fonte: Próprio autor

Na Figura 22, em vermelho encontra-se os dados obtidos através do sistema de medição. A nuvem cinza na parte de traz representa os dados obtidos com treinamento dos dados a partir da matriz de probabilidades encontrada com aplicação cadeia de Markov. É possível observar que os valores obtidos através da simulação seguem o padrão apresentado pelos dados do consumo, indicando que o sistema pode ser modelado utilizando o modelo. Caso o sistema não obedecesse aos princípios das cadeias de Markov, o padrão exibido pelos valores simulados não seguiria o padrão estabelecido pelos dados reais.

A Figura 24 mostra a média dos valores de energia obtidos através da medição em dias úteis em azul e a média dos valores obtidos com simulação em vermelho. É possível observar uma proximidade dos padrões obtidos através do sistema de monitoramento e simulação.

Figura 24 - Média dos valores obtidos com sistema de monitoramento (azul) e simulação (vermelho)

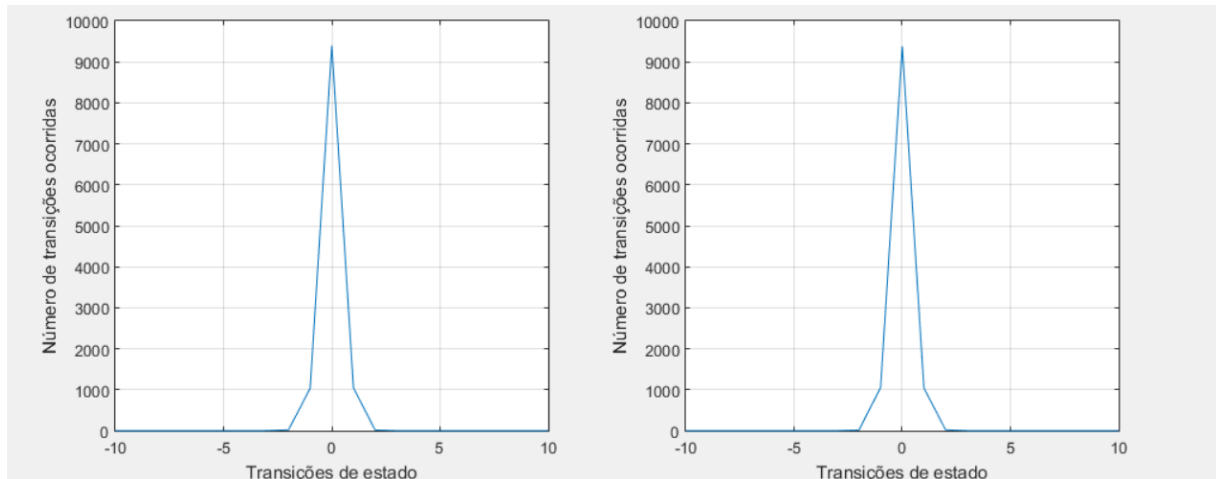


Fonte: Próprio autor

Para a validação do modelo obtido, foi realizada também uma simulação considerando as transações de estado ocorridas entre cada intervalo de hora. A diferença entre o estado do intervalo atual e o estado do próximo intervalo foi acumulada, resultando em um histograma. Tal análise é importante na investigação do comportamento temporal do modelo. A comparação

dos histogramas dos dados obtidos com dados reais (esquerda) e dados simulados (direita) encontra-se na Figura 25.

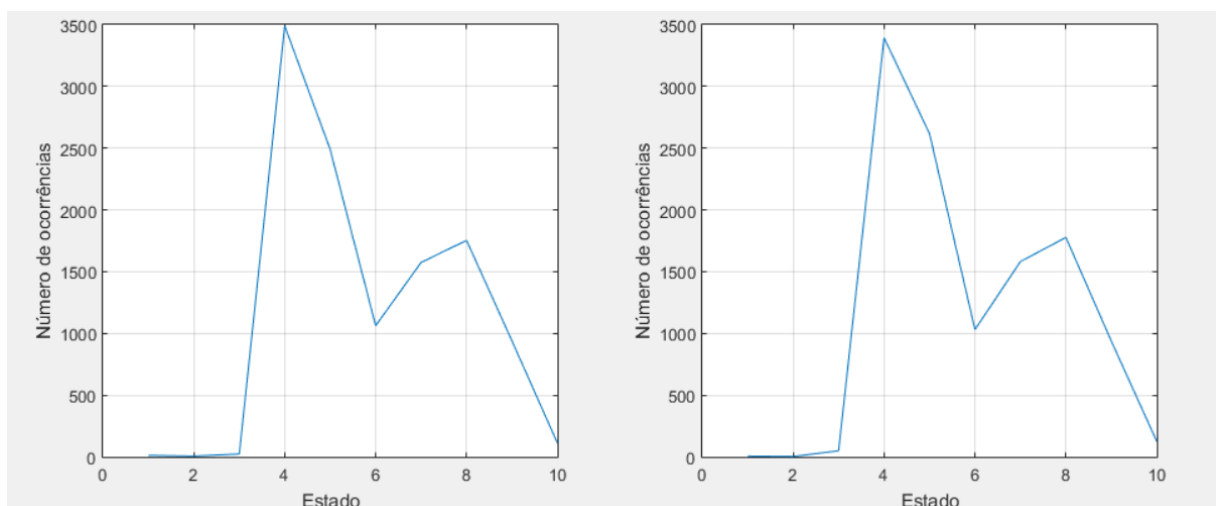
Figura 25 - Comparação de transições de estado entre dados reais (esquerda) e dados simulados (direita)



Fonte: Próprio autor

É possível analisar, a partir da Figura 25, que há uma proximidade considerável entre as transições de estado ocorridas entre os dados obtidos via sistema de monitoramento (esquerda), e dados obtidos através de simulação (direita). A Figura 26 demonstra um histograma comparando o número de ocorrências entre dados reais (esquerda) e dados simulados (direita).

Figura 26 - Histograma de ocorrências de estado com dados reais (esquerda) e dados simulados (direita)



Fonte: Próprio autor

Observando comparação presente na Figura 26, é possível observar a similaridade com que os estados ocorrem nos dados reais (esquerda) e no sistema obtido com dados simulados (direita).

Analisando as Figuras 22, 24, 25 e 26, é possível observar que os dados de consumo de energia ativa obtidos via sistema de monitoramento podem ser modelados via cadeia de Markov, tendo em vista a proximidade dos gráficos obtidos a partir de resultado de simulação e dados reais.

4.2 Transação de energia utilizando criptomoeda

4.2.1 Introdução

A forma como a energia é utilizada vem alterando nos últimos anos. O surgimento de novas formas de carga, como os carros elétricos, e o aumento da produção de energia em formas mais diversas, ocasionado também pelo aumento da produção de energia a partir de fontes renováveis de produção, bem como a ampliação dos estudos que envolvem redes inteligentes, são alguns exemplos deste fato. Por outro lado, a forma como o mercado de compra e venda de energia não obteve mudanças significativas.

A regulamentação da geração distribuída no Brasil em 2012 com a REN 482 permitiu um novo modo de utilização da energia elétrica. Permitiu-se que unidades consumidoras pudessem produzir sua própria energia a partir de fontes renováveis de energia, obedecendo algumas ressalvas estabelecidas pela resolução. O intuito da geração distribuída no Brasil baseia-se na compensação do indivíduo a partir de sua própria produção, não permitindo, assim, a negociação da energia produzida. Portanto, não é possível que um cliente que possua mais espaço disponível que sua necessidade para abater seu consumo total, de produzir e vender seu excedente.

O uso de criptomoedas, que são moedas digitais com técnicas de criptografia para sua validação, vem crescendo consideravelmente nos últimos anos. A utilização dessas moedas virtuais não requer uma autoridade central, sendo a rede de computadores que contem seus dados o agente regulador. Desde a introdução do *blockchain*, tecnologia de criptomoeads mais conhecida e na qual baseia-se o *Bitcoin*, Nakamoto (2008) e outras criptomoedas, o interesse, uso e valor desses ativos vem se expandindo de forma exponencial. O *blockchain* trabalha com registros distribuídos, o que significa que é necessário que 50% mais um dos computadores concordem com uma transação para que seja considerada válida. Além disso, os usuários dessa rede dividem-se em dois: comuns e mineradores. Os usuários comuns são aqueles que participam da rede transacionando valores monetários, enquanto que os mineradores são

aqueles que utilizam parte de seu processamento computacional para validar transações realizadas.

Considerando a revolução recente na produção e consumo de energia elétrica, e o surgimento de novas tecnologias de transação monetária, é razoável associar ambos os conceitos em prol de encontrar um novo modelo de comercialização de energia.

Hahn; Chen (2017) propõem um sistema de leilão de energia utilizando a criptomoeda Ethereum, através de contratos inteligentes, que funciona como um contrato normal, onde suas condições são fixadas através de código. Tendo tais condições obedecidas, a transação do dinheiro segue o fluxo estabelecido também em código. Além da Ethereum, outras moedas também possuem tal funcionalidade, como EOS, NEO e Cardano. Mengelkamp (2018) analisa o caso da micro-rede instalada no Brooklyn, em Nova Iorque, onde os seus usuários conseguem comprar e vender energia localmente utilizando criptomoedas, analisando também se a rede atende a pré-requisitos necessários para um mercado de energia em microrrede.

Winter (2018) fala que o nível de utilização do *blockchain* em redes inteligentes pode ser classificado como de forma fundamental ou melhorada. A forma fundamental implica apenas a utilização da infraestrutura descentralizada do *blockchain* para transferências, enquanto que a forma melhorada implica na utilização de uma plataforma de computação descentralizada disponível no *blockchain*. Esta capacidade computacional poderia promover dinamização das cargas, precificação dinâmica e predição mais assertiva do consumo e produção de energia. Também segundo Winter (2018), quatro são as possibilidades dessa implementação: mercado, medição, conhecimento aplicado e controle.

O caso do mercado é a utilização fundamental do *blockchain*. Através da utilização de uma moeda virtual, uma comunicação P2P é estabelecido e os participantes da rede podem então transacionar energia entre si. Toda a estrutura utilizada é a mesma do modelo tradicional de consumo de energia, tanto o modelo financeiro da comercialização de energia, como os dados e estrutura física.

No modelo envolvendo a medição, além da utilização do *blockchain* para transação financeira, envolve também sua utilização de processamento computacional descentralizado. O incremento comparado ao modelo de mercado se faz presente na utilização de uma infraestrutura de medição comunicando com a plataforma computacional descentralizada, os usuários passam a ter maior conhecimento a respeito do seu consumo de energia, além de fornecer maior segurança em relação aos valores de energia produzidos e consumidos, tornando mais difícil a ocorrência de fraudes.

O modelo de conhecimento aplicado envolve, além da estrutura de medição, o uso de inteligência nos dados presentes na plataforma computacional descentralizada, com a finalidade de potencializar o uso do sistema, com, por exemplo, maior conhecimento a respeito das perdas ocorridas e da previsão de consumo e produção.

O modelo de controle propõe também que o sistema de comercialização utilizando *blockchain* permita a utilização de medição inteligente conectada também aos circuitos de energia menores do usuário, possibilitando um controle interno do consumo de energia ao atuar nesses circuitos conforme perfil de consumo requerido pelo usuário.

Mengelkamp (2018) afirma que criptomoedas baseadas em *blockchain* possuem, entretanto, algumas ressalvas. O *blockchain* é uma tecnologia de registro distribuído, portanto, para que uma transação seja validada, é necessário que metade dos computadores da rede concorde com aquela transição. Assim, quanto mais transições ocorrem, mais o sistema se torna lento, o que evita a escalabilidade da tecnologia.

Outro problema existente na tecnologia do *blockchain* consiste no fato de que as transações ocorridas na cadeia devem ser validadas por usuários conhecidos como mineradores. Para que validar as transações, os mineradores gastam energia elétrica com processamento de seus computadores. Portanto, para que seja viável sua presença na cadeia, os mineradores devem receber uma compensação financeira por parte de quem realiza a transação. Além disso, o gasto de energia existente na validação dos dados vai de encontro à lógica de utilizar a tecnologia como forma de comercializar energia. Como alternativa às criptomoedas baseadas em *blockchain*, surgiu a criptomoeda IOTA.

4.2.2 IOTA

IOTA foi desenvolvida baseada na tecnologia conhecida como *Tangle*. O *Tangle* usa grafos acíclicos dirigidos (DAG, em inglês) como base de seu funcionamento. O IOTA foi desenvolvido com a finalidade de fornecer um meio de transição para a indústria IoT. Essa criptomoeda vem sendo apontada como a evolução do *blockchain*.

O IOTA possui algumas características particulares. As transações do IOTA (TXs) são aprovadas por outras transações, dispensando a participação de mineradores na rede. Portanto, as transferências realizadas com IOTA são gratuitas, independente do volume monetário transacionado. Além disso, seu modo de funcionamento faz com que seja uma tecnologia escalável, pois quanto maior o uso do sistema, mais rapidamente novas transações são aprovadas. Além do envio de valores monetários, uma transação em IOTA permite o envio de

mensagens. Transferências valor nulo são conhecidas como meta-transações. Cada transação deve estar relacionada a um endereço dentro da rede. Como não existem mineradores, o consenso da rede é obtido pelos próprios usuários. A ausência de mineradores faz com que todos os usuários da rede *Tangle* não tenham distinção entre si. Cada nova transação deve aprovar duas transações mais antigas, e transação ainda não aprovadas são conhecidas como *tips*.

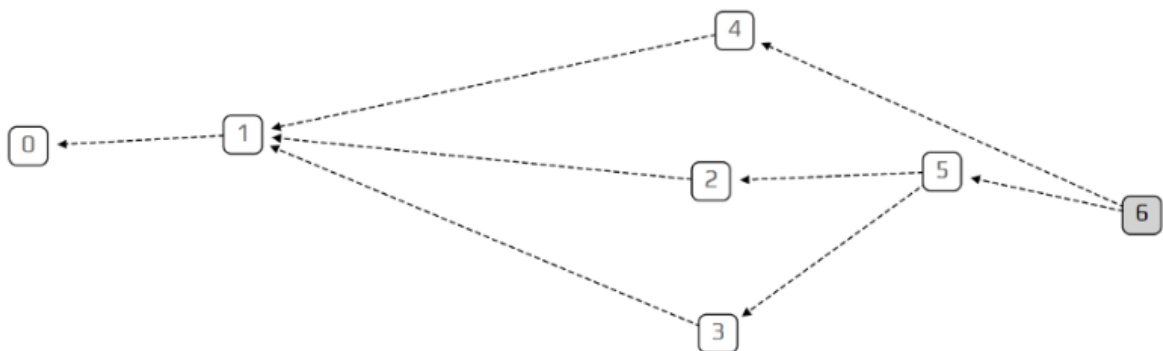
Seeds e endereços também são conceitos importantes para o entendimento do funcionamento do IOTA. Os valores monetários do IOTA são armazenados em carteiras virtuais, e cada uma dessas carteiras é protegida com uma chave privada, conhecida como *Seed*, que é composta por 81 trytes, i.e., qualquer letra maiúscula e o número 9. Como o *Seed* é a única proteção para a carteira, deve ser guardada sob sigilo.

Os endereços, também conhecidos como chaves públicas, são gerados a partir da *seed*. A partir desses endereços é possível que uma carteira receba transações.

A referência de implementação do IOTA (IRI, em inglês) é um *software* em Java que define o protocolo do IOTA. Computadores que processam o *software* IRI são chamados de nós IRI. Dentro das funções dos nós IRI, é válido citar a validação das transações, guardar transações validadas nas carteiras e permitir a interação dos usuários com a IRI.

A Figura 27 demonstra de forma simples o funcionamento da rede *Tangle*.

Figura 27 – Princípio de funcionamento da rede *Tangle*



Fonte: <iota.org>

Como é possível observar na Figura 27, cada vértice da rede representa uma transação. Quando uma nova transação aparece na rede, ela escolhe duas transações mais antigas para serem validadas. Na Figura 21, a transação 5 aprova as transações 2 e 3, por exemplo.

A transação 6 é uma transação que ainda não foi aprovada, embora ela tenha participação nas aprovações das transações 4 e 5.

A aplicação de IOTA se faz interessante em transações máquina-para-máquina (M2M), onde pagamentos de menores valores são mais frequentes. Por exemplo, na montagem de uma indústria, todo o maquinário poderia ser alugado com pagamentos referentes a cada hora de produtividade dos aparelhos. Aplicações para IOTA vêm sendo desenvolvidas também nas áreas de saúde, segurança, automotiva e cadeiras de suprimento e logística. Foi desenvolvido, na Holanda, um sistema de carregamento de veículos elétricos utilizando IOTA.

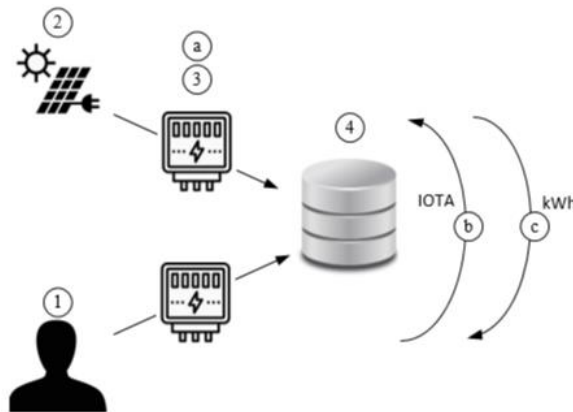
A aplicação da IOTA na transição de energia elétrica se faz interessante pela possibilidade de transacionar a energia a cada kWh produzido.

4.2.3 Sistema proposto

O sistema proposto neste trabalho propõe uma alternativa para o modelo atual de comercialização de energia. Tal modelo baseia-se em modelo similar ao de conhecimento aplicado, proposto por Winter (2018). A única ressalva encontra-se na ausência da possibilidade de utilizar a rede *Tangle* como plataforma de computação descentralizada. O modelo ideal para esse tipo de plataforma encontra-se na utilização de contratos inteligentes, que são contratos estabelecidos através de código respectivo a cada criptomoeda, onde o cumprimento das premissas existentes no contrato resulta em transação automática de valores monetários. Entretanto, o modelo de contrato inteligente da criptomoeda IOTA, conhecido com Qubic, ainda está em fase de desenvolvimento. Sendo assim, foi necessário o desenvolvimento de uma biblioteca em *Ruby* para a utilização do IOTA neste trabalho.

O sistema tem a intenção de, a partir do monitoramento de dados de consumo e produção de eletricidade, promover uma troca virtual de créditos de energia e realizar transações referentes aos valores acordados por IOTA. Os dados de consumo utilizados no sistema foram os mesmos obtidos pelo sistema de monitoramento desenvolvido, enquanto que os dados de consumo foram obtidos a partir de um sistema de monitoramento de uma usina fotovoltaica com potência nominal de 300 kWp. Uma visão geral do sistema proposto encontra-se na Figura 28.

Figura 28 – Sistema proposto



Fonte: Próprio autor

O sistema pode ser dividido em 4 partes:

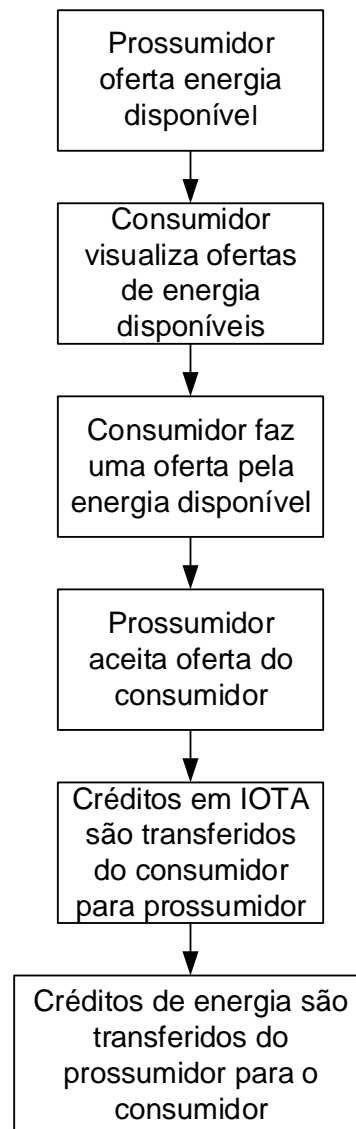
- 1) Monitoramento do consumo de energia elétrica do usuário, para que possa ser mensurada a energia utilizada e conseqüentemente a energia necessária a ser adquirida;
- 2) Monitoramento da produção de energia, de modo que o prossumidor possa monitorar a quantidade de energia disponível para venda;
- 3) Um sistema de monitoramento confiável, de modo a assegurar a veracidade das informações a respeito da energia consumida e da energia produzida;
- 4) Um sistema de transição com banco de dados confiável, a prova de usuários maliciosos, onde a comercialização de energia possa ser feita de maneira segura. No estado da arte desse modelo, esse banco de dados deverá ser conectado ao sistema de faturamento da distribuidora, de modo a assegurar que a energia comprada pelo cliente não seja tarifada duas vezes.

Como pode ser observado também na Figura 21, o funcionamento do sistema pode ser assim descrito:

- a) Prossumidor oferece sua energia disponível no sistema;
- b) Consumidor ver a oferta e faz um lance, informando a quantidade de energia e preço disposto a pagar por aquela oferta. Consumidor aceita a oferta e o sistema transfere o equivalente em IOTA da transação acordada;
- c) Após a transação monetária, o sistema verifica a veracidade da transação, e após a validação, envia os créditos de energia;

Um fluxograma do processo de comercialização de energia se encontra na Figura 29.

Figura 29 - Fluxograma do processo de comercialização de energia desenvolvido



Fonte: Próprio autor

4.2.4 Resultados

O sistema foi testado na mesma aplicação do sistema de monitoramento, portanto, também foi desenvolvido utilizando RoR. Alguns códigos na linguagem de programação NodeJS também foram desenvolvidos, devido à ausência de uma biblioteca de interação entre o IOTA e o Ruby.

Além de mostrar os dados de consumo, o sistema possui uma biblioteca de *web scrapping* chamada Watir, capaz de acessar dados de outras páginas. Tal aplicação foi necessária para a extração dos dados de geração da usina monitorada, com dados constando em outra

página na Internet. O sistema tem a responsabilidade também do que seria da distribuidora de energia, em um estado da arte, de verificação e confirmação das transições.

A rede IOTA fornece uma de desenvolvimento, onde é possível realizar testes e transação de dinheiro sem valor. Esta rede foi utilizada no desenvolvimento deste trabalho.

Quando o usuário acessa o sistema, é realizado um cadastro, onde afirma se há ou não produção de energia no local. É necessário também o preenchimento de dados referentes à carteira virtual, *seed* e endereços, que realizará as transações em IOTA.

A partir daí, é possível visualizar, na página do prossumidor, a quantidade de energia disponível para a venda, conforme Figura 30.

O prossumidor confirma então a quantidade de energia disponível para venda e o preço por unidade de kWh, e a torna disponível de ser visualizada por outros usuários. O consumidor pode, a partir daí, visualizar as ofertas de energia disponíveis no sistema, conforme Figura 31.

Figura 30 – Quantidade de energia disponível para a venda

Energy available	Price	Available?
117.0	0.6	NOT

Fonte: Próprio autor

Figura 31 – Energia disponível para compra

ID	UC	Offer (kWh)	Price (\$/kWh)
1	101010	117.0	0.6

Fonte: Próprio autor

O consumidor pode então fazer uma oferta de compra pela energia, conforme Figura 32.

Figura 32 – Página de finalização da oferta

Make an offer

Seller: 101010
Kwh available: 117.0
Price (R\$/kWh): 0.6

Details:
Energy ID: 1

kWh amount desired:

Fonte: Próprio autor

Quando o usuário entra na página de compra, é possível visualizar o saldo de sua carteira e o valor respectivo em reais.

O sistema mostra ao possível comprador detalhes da oferta, e a oferta pode ser feita. Algumas prevenções de erros foram desenvolvidas no sistema, como a impossibilidade de comprar mais energia do que a disponível ou de serem inseridos valores negativos.

Depois que a oferta é feita, é mostrada ao respectivo prosumidor, junto com todas as ofertas feitas por aquela energia. Quando o vendedor aceita a oferta, o sistema automaticamente rejeita todas as outras.

O sistema calcula o montante total necessário a ser transferido e tenta transferir o valor do comprador para o vendedor. Quando começa o processo de transferência do dinheiro, o uma *tag* contendo 27 tentativas aleatórias é criada. À medida que a transferência é concluída, o sistema verifica, pela hash da transação, se o valor enviado é igual ao custo de energia, se o endereço do receptor da transação é o mesmo do usuário prosumidor e se a *tag* aleatória gerada é a mesma na transação como no sistema. Ao validar essas três características, o crédito de energia é enviado ao comprador e o crédito ao consumidor é reduzido.

Para validar o uso do sistema, foi feita uma simulação. Nesse caso, o usuário fez uma oferta para comprar 50 kWh, custando 0,6 IOTA/kWh. Depois que o vendedor aprovou a oferta, 30 IOTA foram transferidos para o vendedor. A Figura 33 mostra o saldo de usuários antes e depois da compra diretamente da carteira virtual *Trinity*.

Figura 33 – Comparação de balanço das carteiras de IOTA

Energy Seller 476 i	Energy Seller 506 i
Energy Buyer 293 i	Energy Buyer 263 i

Fonte: Próprio autor

4.3 Componentes de um mercado de microrrede de energia

Baseado em (MENGELKAMP, 2018), sete fatores para uma operação eficiente de um mercado de energia em microrrede são apresentados.

4.3.1 Configuração da rede (C1)

É necessária uma definição clara dos participantes, da forma como a energia é transacionada e qual tipo de energia é transacionada.

Com relação aos participantes, a rede é composta basicamente por consumidores e prossumidores, que deverão comprar e vender energia, respectivamente. Prossumidores podem ser eventuais compradores de energia. Os participantes também devem ser aqueles pertencentes a comunidade da microrrede, em caso de implementação do modelo totalmente descentralizado, ou pré-estabelecidos em caso contrário.

A forma como a energia é transacionada deve ser observada para a configuração da rede, se em forma física através de uma microrrede construída ou da rede de distribuição já existente.

Além do mais, deve-se considerar também se apenas eletricidade é comercializada, ou também outros tipos de energia, como térmica.

4.3.2 Conexão da rede (C2)

O número de conexões da microrrede com a rede de distribuição é um fator a ser considerado no desenvolvimento da rede, principalmente para o cálculo de oferta e demanda de energia elétrica da microrrede e da rede de distribuição.

Existe uma grande diferença entre os participantes da rede que estão fisicamente conectados, como em uma rede totalmente descentralizada, e dos usuários que estão apenas virtualmente conectados à rede de comercialização de energia, pois estes não podem ser

fisicamente desacoplados da rede, enquanto que microrredes físicas podem acoplar e desacoplar da rede de distribuição em caso de necessidade ou risco. A capacidade de armazenamento e produção de energia são fatores essenciais para a definição desse fator.

4.3.3 Sistema de informação (C3)

Um sistema de informação eficiente é necessário para conectar todos os participantes da rede e ser responsável pelas comercializações de energia ocorridas. O sistema deve ser confiável, de tratamento igual para todos os usuários e eficiente.

A utilização de contratos inteligentes é uma solução ideal para tal funcionalidade.

A segurança das transações também deve ser levada em consideração no desenvolvimento do sistema, de modo a evitar usuários maliciosos venham a cometer fraudes. A utilização de medidores de energia inteligentes também é necessária, de modo que os usuários possam monitorar sua produção e consumo de energia, e efetivar a transação de energia de um modo mais eficiente.

É de responsabilidade também do sistema informativo a validação das transações, de modo a verificar se o valor correspondente enviado pelo comprador teve o destino certo, bem como seu valor. Além disso, deve verificar também a transação da energia para o consumidor.

4.3.4 Mecanismo de mercado (C4)

O mecanismo de mercado consiste na forma e nas regras de pagamento, e esclarece todo o sistema de ofertas, lances e transação de energia. Seu principal objetivo é promover a transação de energia do modo mais eficiente, encontrando a melhor solução para compradores e vendedores. É de interesse do sistema que esse mecanismo atue em tempo real.

O tempo de comercialização é também um fator importante em seu desenvolvimento. Diferentes horizontes de tempo nas transações ocasionam diferentes características de mercado, como por exemplo, um sistema baseado em leilões, as transações podem ocorrer de forma diária, enquanto que transações P2P podem ser realizadas em tempo real.

Ambas situações também podem resultar em transações de longo prazo, com contratos com maior duração de tempo.

4.3.5 Mecanismo de precificação (C5)

A definição dos preços nesses sistemas baseia-se essencialmente na oferta e demanda de energia presente. O preço da energia pode ser encarado como um sinal de excesso de energia, em caso de redução dos preços, ou de escassez, quando o preço da energia se elevar.

Da mesma forma que a energia ofertada no sistema de distribuição, deve incidir também uma série de taxas e impostos sobre a energia comercializada nesse tipo de sistema.

A precificação também pode levar em consideração características do consumidor, como distância do local de produção e se o mesmo se encontra fisicamente ou apenas virtualmente conectado ao sistema, bem como características do prossumidor, como quantidade de energia disponível e fonte primária da energia produzida.

4.3.6 Sistema de gerenciamento de comercialização da energia (SGCE)

O objetivo do SGCE é promover o fornecimento correto da energia para o usuário do sistema e gerenciar a compra ou venda da energia. Tal sistema necessita da leitura em tempo real dos dados de consumo e geração da unidade, podendo, assim, prever a capacidade de produção de energia ou necessidade de compra de energia em um determinado período. Com tais informações, esse sistema é capaz de gerenciar todo o consumo ou produção do sistema, bem como garantir o fornecimento e comercialização em um período futuro. Um sistema eficiente deve sempre comprar energia quando o preço da energia cair, e oferecer energia quando o preço da energia se elevar.

O SGCE também deve ter acesso a dados das contas de criptomoeda do usuário, de modo a ser hábil a realizar transações em tempo real, assim que as condições pré-estabelecidas forem definidas.

4.3.7 Regulação

A regulação é responsável por determinar as políticas de funcionamento do mercado de transação de energia. Assim, determinam as regras do mercado, o que é permitido, quais taxas incidem e como o sistema deve integrar a rede de distribuição já existente. A regulação tem um papel essencial na viabilidade do sistema, podendo elaborar políticas, juntamente com o governo, que contribuam para o desenvolvimento de tal sistema, como subsídios e isenções. Caso seja percebido um movimento negativo na implementação desses sistemas, pode também

desincentivar tais organizações. Para a implementação na vida real, a regulação é o fator mais importante.

4.4 Avaliação dos fatores no sistema desenvolvido

Com relação ao fator C1, o sistema desenvolvido neste trabalho é claro quanto a sua composição. Consumidores e prossumidores são estabelecidos, e qualquer consumidor pode tornar-se um prossumidor, e vice-versa, sem distinção entre usuários. No caso prático, apenas um consumidor e um prossumidor fizeram parte do sistema, entretanto sua escalabilidade é viável, dependendo apenas da instalação do sistema de monitoramento em novas unidades.

A transação de energia implementada foi realizada apenas de forma virtual, tendo sido utilizada a rede de distribuição para o efetivo consumo e produção da energia. Além disso, apenas eletricidade é comercializada no sistema proposto.

O fator C2 não se aplica ao sistema desenvolvido, tendo em vista que apenas a rede de distribuição local foi utilizada. Portanto, nesse caso, a preocupação com relação a oferta e demanda de energia é de responsabilidade da distribuidora do ponto de vista de estabilidade do sistema, enquanto que tais aspectos são úteis aos participantes do sistema apenas ao considerar a energia a ser comercializada.

Considerando o fator C3, o sistema desenvolvido possui um sistema de informação, onde conecta todos os usuários e é responsável pelas transações de energia ocorridas. O sistema é confiável, e possui tratamento isonômico entre todos os usuários. Por utilizar uma moeda de criação recente, ainda em desenvolvimento, não foi possível utilizar contratos inteligentes para a implementação deste fator. Alguns métodos de segurança foram implementados, de modo a validar as transações e evitar fraudes nas transações de valores e de créditos de energia.

O sistema desenvolvido possui um mecanismo de mercado (C4) bem desenvolvido, ficando claro para o usuário onde deve ofertar sua energia disponível, no caso dos prossumidores, e onde procurar por energia disponível para compra, no caso dos consumidores. Os prossumidores estabelecem a quantidade de energia em kWh disponível para venda e o respectivo preço por unidade, e é de responsabilidade do consumidor fazer uma oferta, caso haja interesse. O sistema é parecido com um leilão, onde a melhor oferta para o prossumidor deve ser aceita. O sistema funciona também em tempo real, sendo as transações de energia e IOTA realizados no momento em que o prossumidor aceitar a oferta. O tempo de comercialização depende do prossumidor, tendo em vista que este pode levar o tempo que considerar necessário para aceitar a oferta de compra de energia.

O mecanismo de precificação (C5) da energia comercializada no sistema baseia-se exclusivamente no preço fornecido pelo prosumidor da sua energia disponível, sendo ele o único responsável por disponibilizar o preço da energia a ser transacionada. O sistema tende, assim, a obedecer a oferta e demanda de energia disponível.

Um estudo mais aprofundado deve ser feito para verificar os efeitos contábeis e fiscais do sistema proposto, taxas e impostos a serem incididos em um eventual sistema implementado.

O sistema proposto possui um dispositivo semelhante a um SGCE (C6), capaz de monitorar a produção e consumo de energia dos usuários em tempo real, sendo capaz de prever o consumo de energia da unidade nos próximos 30 dias. Entretanto, não foi implementado no dispositivo uma inteligência capaz de realizar, automaticamente, ofertas de compra e venda de energia dependendo das necessidades do usuário.

A regulação (C7) é o fator de maior impedição para a real implementação e escalabilidade do sistema. Atualmente, a utilização de criptomoedas para realizar transação de energia necessita de maior estudo para a identificação de suas características de uso por parte dos órgãos regulatórios.

A Tabela 5 resume todos os fatores, bem como a conclusão de sua execução.

Tabela 5 – Resumo dos fatores e sua execução no sistema proposto

Fator	Execução	Comentário
C1	Aplicado	Características do sistema bem definidas
C2	Não executado	Sistema utiliza rede de distribuição existente
C3	Aplicado	Sistema de informação implementado
C4	Parcialmente aplicado	Mecanismo de mercado implementado, mas de forma simples, necessitando de melhorias do ponto de vista mercadológico
C5	Parcialmente aplicado	Mecanismo de precificação simples implementado, necessitando maior estudo do ponto de vista contábil e econômico
C6	Parcialmente aplicado	Sistema de gerenciamento capaz de monitorar consumo e geração implementado, mas carece de inteligência de transação automática
C7	Não executado	Regulação ainda não permite a implementação de tais sistemas

Fonte: Próprio autor

4.5 Conclusão

Neste capítulo, inicialmente foi feita uma revisão a respeito de modelagem de perfil de consumo, e uma introdução teórica sobre cadeia de Markov foi introduzida. O processo de modelagem foi explicado, e a simulação inicial desenvolvida com MATLAB foi apresentada.

Em seguida, o conceito envolvendo criptomoedas foi abordado, bem como alguma de suas funcionalidades, tecnologias e vantagens e desvantagens respectivas. Os conceitos de IOTA *Tangle* foram introduzidos, bem como um sistema proposto para o desenvolvimento de transação de energia através de criptomoedas. O uso do sistema proposto foi demonstrado, e uma transação de energia utilizando a rede de desenvolvimento do IOTA foi realizada.

5 CONCLUSÕES PARCIAIS

Neste trabalho foi apresentado um sistema de monitoramento de consumo de energia elétrica para unidades conectadas em média tensão, que tem como objetivo mostrar dados referentes ao consumo de energia elétrica das unidades na qual for instalado, bem como propor melhorias com base no perfil de consumo.

O desenvolvimento do protótipo foi detalhado, e sua funcionalidade foi apresentada. O dispositivo instalado na FUNTELC foi apresentado, podendo ser visualizado todos os dados presentes na plataforma criada, a qual foi desenvolvida utilizando a linguagem de programação *Ruby*, com auxílio do *framework Ruby on Rails*.

O sistema detectou alguns parâmetros relevantes do consumo da FUNTELC, como dados de consumo de energia ativa e reativa, demanda e fator de potência, divididos por hora, dia e mês, bem como um elevado consumo no período da madrugada e uma possível redução de demanda contratada, podendo obter, assim, uma redução de até 10% do valor final da conta de energia.

Foram demonstradas também algumas aplicações possíveis para o sistema de monitoramento. Inicialmente, foi demonstrada a possibilidade de, com base nos dados de consumo adquiridos, realizar uma modelagem utilizando cadeias de Markov do perfil de consumo de energia elétrica da unidade. Algumas simulações foram desenvolvidas utilizando MATLAB, e comparações entre dados reais e simulados avaliando transição de estados, frequência de estados e média foram realizados, comprovando que o sistema pode ser modelado como uma cadeia de Markov e que a modelagem de fato estava coerente.

Em seguida, foi apresentado um modelo de transação de energia utilizando criptomoeda. Devido as vantagens em comparação com outras criptomoedas, como ausência de mineradores e velocidade de transição, a simulação foi feita utilizando IOTA, que utiliza a tecnologia *Tangle* em seu funcionamento. Um monitoramento de uma usina fotovoltaica foi realizado, e tal unidade foi utilizada como modelo de prosumidor, enquanto que a FUNTELC foi utilizada para a simulação de um consumidor. Uma simulação foi realizada, onde a FUNTELC comprou créditos virtuais de energia da usina fotovoltaica. A rede de desenvolvimento da IOTA foi utilizada.

Os sete fatores de (MEGELKAMP, 2017) para a existência de um mercado de energia utilizando criptomoedas foram analisadas para o sistema desenvolvido. Verificou-se a existência de dois fatores no modelo desenvolvido, outros três fatores sendo parcialmente aplicados e outros dois fatores não executados no trabalho desenvolvido.

5.1 Atividades futuras

Como proposta para trabalhos futuros, sugere-se a utilização de modelos de controle de carga utilizando o sistema de monitoramento desenvolvido, de modo que o sistema possa também atuar diretamente no controle do consumo, demanda e fator de potência da unidade, resultando em maiores impactos positivos no consumo de energia do cliente.

É interessante também que haja uma aplicação do sistema de cadeias de Markov em uma unidade consumidora onde haja a presença de geração distribuída, de modo a verificar se tal sistema se aplica em uma cadeia de Markov.

O desenvolvimento de um modelo de comercialização de energia totalmente descentralizado, com uma rede isolada e presença de sistemas de armazenamento de energia também é uma proposta interessante a ser desenvolvida a partir do desenvolvimento deste trabalho.

Por fim, sugere-se também que, ao ser lançado o Qubic, o sistema de contratos inteligentes utilizando IOTA, que seja utilizado em sistemas de comercialização de energia utilizando IOTA.

REFERÊNCIAS

- WINTER, T.M.G.L. The Advantages and Challenges of the Blockchain for Smart Grids. Dissertação (Mestrado em Engenharia Elétrica). Delft University of Technology, Holanda, 2018.
- MEGELKAMP, E. *et al.* Designing microgrid energy markets. A case study: The Brooklyn Microgrid. *Applied Energy*, Volume 210, 2018, páginas 870-880, ISSN 0306-2619.
- NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Disponível em <bitcoin.org>. Acesso em 03 de março de 2019.
- HAHN A.; CHEN S., Smart Contract-based Campus Demonstration of Decentralized Transactive Energy Auctions. *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, 2017, páginas 1-5.
- UHRIG, M.; MUELLER, R.; LEIBFRIED, T. Statistical consumer modelling based on smart meter measurement data. *2014 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, Durham, 2014, pp. 1-6.
- GROß, D.; WIEST, P.; RUDION, K. "Comparison of stochastic load profile modeling approaches for low voltage residential consumers," *2017 IEEE Manchester PowerTech*, Manchester, 2017, páginas 1-6.
- SHEKIN, Theodore J. Markov chains and decision processes for engineers and managers. Taylor & Francis Group, LLC. 2011.
- LABEEUW W.; DECONINCK G. "Residential Electrical Load Model Based on Mixture Model Clustering and Markov Models," in *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1561-1569, Agosto, 2013.
- RAI, P.; REHMAN, M. "ESP32 Based Smart Surveillance System," *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, Pakistan, 2019, pp. 1-3.
- SARJERAO, B. S.; PRAKASARAO A., "A Low Cost Smart Pollution Measurement System Using REST API and ESP32," *2018 3rd International Conference for Convergence in Technology (I2CT)*, Pune, 2018, pp. 1-5.
- ALLAFI, I.; IQBAL, T. "Design and implementation of a low cost web server using ESP32 for real-time photovoltaic system monitoring," *2017 IEEE Electrical Power and Energy Conference (EPEC)*, Saskatoon, SK, 2017, pp. 1-5.

- BISWAS, S. B.; IQBAL, M. T., "Solar Water Pumping System Control Using a Low Cost ESP32 Microcontroller," *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, Quebec City, QC, 2018, pp. 1-5.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 14519:2011. Medidores eletrônicos de energia elétrica – Especificação. 2011.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 14520:2011. Medidores eletrônicos de energia elétrica – Métodos de ensaio. 2011.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 14522:2008. Intercâmbio de informações para sistemas de medição de energia elétrica. 2008.
- SUN, Y.; WANG, Z., "The energy consumption monitoring platform design for large-scale industry users based on the GPRS," *2011 Second International Conference on Mechanic Automation and Control Engineering*, Hohhot, 2011, pp. 7827-7830.
- MUMTAZ, F., "Real-Time, PLC based, Energy Monitoring System," *2015 Annual IEEE India Conference (INDICON)*, New Delhi, 2015, pp. 1-5.
- HAN, J. *et al.* "Smart home energy management system including renewable energy based on ZigBee and PLC," in *IEEE Transactions on Consumer Electronics*, vol. 60, no. 2, pp. 198-202, May 2014.
- IRMAK, E., KÖSE, A.; GÖÇMEN, G. "Simulation and ZigBee based wireless monitoring of the amount of consumed energy at smart homes," *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*, Birmingham, 2016, pp. 1019-1023.
- SAHOO, U. K.; PATNAIK, B. "Design and implementation of remote monitoring system of solar lanterns, based on lorawan and cloud technology," *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, 2017, pp. 129-133.
- SUN, Y.; WANG, Z. "The energy consumption monitoring platform design for large-scale industry users based on the GPRS," *2011 Second International Conference on Mechanic Automation and Control Engineering*, Hohhot, 2011, pp. 7827-7830.
- PEREIRA, R. I. S. P. Rede de monitoramento IOT aplicada em plantas de microgeração fotovoltaica. Tese (Doutorado em Engenharia Elétrica). Universidade Federal do Ceará. 2018.
- DUPONT, I. M. Sistema embarcado Linux para monitoramento em nuvem aplicado a bombeamento fotovoltaico. Dissertação (Mestrado em Engenharia Elétrica). Universidade Federal do Ceará, 2017.

WANG, X. *et al.* "Development of auxiliary line loss management system in Hebei grid," *45th International Universities Power Engineering Conference UPEC2010*, Cardiff, Wales, 2010, pp. 1-4.

BARAI, G. R.; KRISHNAN S.; VENKATESH, B., "Smart metering and functionalities of smart meters in smart grid - a review," *2015 IEEE Electrical Power and Energy Conference (EPEC)*, London, ON, 2015, pp. 138-145.

NIU, S. *et al.* (2013). "Electricity consumption and human development level: A comparative analysis based on panel data for 50 countries". *International Journal of Electrical Power & Energy Systems*. 53. pp. 338–347.

QUEDRAOGO, N. Energy consumption and human development: evidence from a panel cointegration and error correction model. 2013. Disponível em:

<https://www.researchgate.net/publication/258932928_Energy_Consumption_And_Human_Development_Evidence_From_A_Panel_Cointegration_And_Error_Correction_Model>.

Acesso em: 29 mar. 2017.

ANEXOS

A formação dos dados é binária, detalhada da seguinte forma:

1° byte:

- *Bit 0* ao *Bit 7*: Número de segundos até o fim do intervalo de demanda ativa atual (LSB);

2° byte:

- *Bit 0* ao *bit 3*: Número de segundos até o fim do intervalo de demanda ativa atual (MSB);

- *Bit 4*: Indicador de reposição de demanda. É complementado a cada reposição de demanda, ou seja, a cada intervalo de 15 minutos com o início de um novo intervalo;

- *Bit 5*: Indicador de intervalo de reativo. É complementado a cada fim de intervalo de consumo de reativo, ou seja, é complementado a cada intervalo de uma hora;

- *Bit 6*: Se igual a 1, indica que os pulsos de energia reativa capacitiva estão sendo computados para cálculo de UFER e DMCR, ou seja, permanece 1 dentro do intervalo de 6 horas estabelecidos dentro de 23h e 7h, estabelecido por cada distribuidora;

- *Bit 7*: Se igual a 1, indica que os pulsos de energia reativa indutiva estão sendo computados para cálculo de UFER E DMCR, complementar, portanto, ao *bit 6* deste mesmo *byte*.

3° byte:

- *Bit 0* a *bit 3*: Segmento horo-sazonal atual:

0001 – ponta;

0010 – fora da ponta;

1000 – reservado;

- *Bit 4* a *bit 5*: Tipo de tarifa:

00 – Azul;

01 – Verde;

10 – Irrigantes;

11 – Outras;

- *Bit 6*: Não usado;

- *Bit 7*: Se igual a 1, tarifa de reativos ativada;

4° byte:

- *Bit 0 a 7*: Número de pulsos de energia ativa desde o início do intervalo de demanda ativa atual (LSB);

5° byte:

- *Bit 0 a 6*: Número de pulsos de energia ativa desde o início do intervalo de demanda ativa atual (MSB);

- *Bit 7*: Não utilizado;

6° byte:

- *Bit 0 a 7*: Número de pulsos de energia reativa desde o início do intervalo de demanda ativa atual (LSB);

7° byte:

- *Bit 0 a 6*: Número de pulsos de energia ativa desde o início do intervalo de demanda ativa atual (MSB);

- *Bit 7*: Não utilizado;

8° byte:

- *Bit 0 a 7*: Complemento do “ou exclusive” dos octetos anteriores;

APÊNDICE

Apêndice A

Código do MATLAB para simulação de cadeias de Markov

Código 1 – dados3.m (Leitura, tratamento inicial dos dados, geração da matriz de probabilidades e treinamento da matriz gerada)

```

clear all
load('C:\Users\Arthur\Documents\UFC\Mestrado\Dissertação\Markov\hc7.mat')
load('C:\Users\Arthur\Documents\UFC\Mestrado\Dissertação\Markov\ea7.mat')
load('C:\Users\Arthur\Documents\UFC\Mestrado\Dissertação\Markov\energial.mat')

x = 0;
eat = (unnamed(96*(x+1):96*(x+1)+95));
hct = (hc7(96*(x+1):96*(x+1)+95));
plot(hct,eat);

max_pro_tag = 0;
array_dividido_por_setor = zeros(1000,96);
ea7_temp = sort(unnamed,2); %organizo pela data
aktuell = ea7_temp(1,2);
last_index_change = 0;
array = -1;
new_ea7 = 0;

%nova normalização
dados_normalizados = (unnamed1(:,1)-
min(unnamed1(:,1)))/(max(unnamed1(:,1))-min(unnamed1(:,1)));
dados_normalizados(:,2)=unnamed1(:,2);

%max_eat = max(eat);
%norm_eat = eat/max_eat;
%norm_eat_alle = ea7/max(ea7);

%número de intervalos de estados
n = 10;
k = 0:1/n:1;
p = zeros(n,n,96);

for h=0:1:95 %coloquei 95 pq vou considerar 96 transações.
    for i=1:1:length(dados_normalizados)-1
        if(dados_normalizados(i,2)==h)
            estado_atual = which_state(n,dados_normalizados(i,1));
            prox_estado = which_state(n,dados_normalizados(i+1,1));
            p(estado_atual, prox_estado, h+1) = p(estado_atual,
prox_estado, h+1) + 1;
        end
    end
end
end

```

```

for h=1:1:96
    for i=1:1:n
        if sum(p(i, :, h)) ~= 0
            p(i, :, h) = p(i, :, h)/sum(p(i, :, h));
        end
    end
end

%initial_state = which_state(n, rand());

%extraí os dados a partir da matriz encontrada

markov_finall = zeros(1000, 350400);

for j=1:1:length(markov_finall(:, 1))
    for h=1:1:length(markov_finall(1, :))
        if h==1
            j
            rand_entre_30_e_40 = rand();
            current_state = which_state(n, rand_entre_30_e_40);
            markov_finall(j, 1) = current_state;

            next_state = next_state_fun(p(current_state, :, h));
            current_state = next_state;
            markov_finall(j, h+1) = current_state;
        else
            r = rem(h, 96);
            if r == 0;
                r = 96;
            end
            next_state = next_state_fun(p(current_state, :, r));
            markov_finall(j, h+1) = next_state;
            current_state = next_state;
        end
    end
end

%20 segundos por 10 anos

interval = 0;
markov_final_test = zeros(96, 1);
for h=(interval*96+1):1:(interval*96+96)
    r = rem(h, 96);
    if r == 0;
        r = 96;
    end
    markov_final_test(r) = markov_finall(h);
end

markov_finall(96);
%markov_final_final(interval*96+1);
%markov_final_final(interval*96+96);

```

```
plot(1:1:96,markov_final_test)
ylabel('Perfil de consumo')
xlabel('Horário')
%axis([0 96 0 110])
axis([0 96 0 11])
```

Código 2 – chart_markov.m (Gerar gráficos)

```
load('C:\Users\Arthur\Documents\UFC\Mestrado\Dissertação\Markov\teste.mat')
user = 939;
interval = 3510;
markov_final_test = zeros(96,1);
markov_final_test2 = zeros(96,1);
for h=(interval*96+1):1:(interval*96+96)
    r = rem(h,96);
    if r == 0;
        r = 96;
    end
    markov_final_test(r) = markov_final1(user,h);
    markov_final_test2(r) = markov_final1(user+1,h);
end

markov_final1(96);

k=.8;

plot(1:1:96,markov_final_test, 'Color',[0 0 0]+k)
hold on
plot(1:1:96,markov_final_test2, 'Color',[0 0 0]+k)
ylabel('Perfil de consumo')
xlabel('Horário')
%axis([0 96 0 110])
axis([0 96 0 11])
```

Código 3 – nuvem.m (Gerar gráfico de nuvem)

```

%load('C:\Users\Arthur\Documents\UFC\Mestrado\Dissertação\Markov\energial.m
at')

meses = 4;
dias = 30;
horas = 96;
total_intervalos = meses*dias*horas; %número total de dados da nuvem
total_dados = 351400; %total de pontos gerados por unidade
inicio = total_dados - 1000 - total_intervalos; %inicio dos últimos 6 meses
total_usuarios = 1000; %número de usuários simulados na nuvem
nuvem_final = zeros(total_intervalos, total_usuarios);
dados_originais = zeros(total_intervalos, 1);
for j=1:1:total_usuarios
    i = 1;
    for h=inicio:1:(total_intervalos+inicio-1)
        nuvem_final(i,j) = markov_final1(j, h)    ;
        i=i+1;
    end
end

k=.8;

for j = 1:1:total_usuarios
    plot(1:1:total_intervalos,nuvem_final(:,j), 'Color',[0 0 0]+k)
    hold on
end

plot(1:1:total_intervalos,round(dados_normalizados(:,1)*10), 'Color',[1 0
0], 'LineWidth',1)

ylabel('Perfil de consumo')
xlabel('Horário')
%axis([0 96 0 110])
axis([0 total_intervalos 0 11])

```

Apêndice B

Código ESP32

```

#include <WiFi.h>
#include <ArduinoOTA.h> //lib do ArduinoOTA
#include <ESPmDNS.h> //lib necessária para comunicação network
#include <WiFiUdp.h> //lib necessária para comunicação network
#include <HTTPClient.h>
#include <rom/rtc.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include <DNSServer.h>
#include <WebServer.h>
#include <WiFiManager.h>

#define uS_TO_S_FACTOR 1000000 /* Conversion factor for micro seconds to seconds */
#define TIME_TO_SLEEP 30 /* Time ESP32 will go to sleep (in seconds) */
#define SERIAL1_RXPIN 12
#define SERIAL1_TXPIN 13

TaskHandle_t Task1;

//HardwareSerial Serial2(2);
HardwareSerial Serial1(2);

char frameMedidor[5000], frameBackupMedidor[300], correctFrameMedidor[300];
//char q1Backup[2880], q2Backup[2880], q3Backup[2880], q4Backup[2880],
q5Backup[2880], q6Backup[2880], q7Backup[2880], q8Backup[2880], crcBackup[2880];
char q1Backup[10], q2Backup[10], q3Backup[10], q4Backup[10], q5Backup[10],
q6Backup[10], q7Backup[10], q8Backup[10], crcBackup[10];
String arrayBackup[10] = "0"; //salva ANO+MES+DIA+HORACOUNTER
String dia_string, mes_string, ano_string, hora_counter_string;
char dia, mes, ano, hora_counter;
String dia2, mes2, ano2, hora_counter2, status_esp, index_data;
int hora_counter_int;
int backup[96][3];

int lastTime;
//Salvar ultimos dados
unsigned long timeFuture, timeSend, timeSendAlert;
int interval = 100;
int temporestantembs, temporestantelsb, temporestante;
int i = 0, k = 0, j = 0;
int n;
int user_id = 2;
String stringOne = "http://104.131.45.58/dados/get_save.json?unidade_id=2&user_id=2";
String stringToBeSent;

```



```

String alertString = "http://104.131.45.58/dados/alerta.json?unidade_id=2&user_id=";
//String alertTwo = "http://104.131.45.58/dados/alert.json?type=2"; //avisa que o esp32 foi
ligado nesse momento
bool incrementar, var_incrementar;
bool start_enviado = false;
bool lost = false;
char all_xor;
String payload2, payload3;
int check_sum_test = 0, nao_envia_test;
bool tratamento = false;
bool flag_nao_envia = false; //avisa e reinicia quando o sistema passar 2 min sem enviar;
unsigned long nao_le = 0;
int reconfig_index;
int var_send_test = 30;
int bug_http = 0;
hw_timer_t *timer = NULL; //faz o controle do temporizador (interrupção por tempo
bool flag_enviando;
int connection_attempts = 0;
int array_backup_index = 0;
char backupData[10]; //Arquivo de backup a ser enviado
bool enviando_backup = false;
char checksum;

HTTPClient http, http1;
WiFiManager wifiManager;

void readReset(void) {

    http1.begin("http://104.131.45.58/reboot.json?unidade_id=2");
    int httpCode3 = http1.GET();
    if (httpCode3 > 0) {
        payload3 = http1.getString();
        Serial.println(httpCode3);
        Serial.println(payload3);
    } else {
        Serial.println("Error on HTTP request");
        bug_http++;
        if (bug_http == 3) {
            // Serial1.end();
            // ESP.restart();
            bug_http = 0;
            connectToWifi();
        }
    }

}

status_esp = String(payload3[10]);
dia2 = String(payload3[19]) + String(payload3[20]);
mes2 = String(payload3[30]) + String(payload3[31]);

```

```

ano2 = String(payload3[40]) + String(payload3[41]) + String(payload3[42]) +
String(payload3[43]);
//COMENTEI AS LINHAS ABAIXO PRA ELE NÃO DEPENDER DA CONEXÃO
//hora_counter2 = String(payload2[53]) + String(payload2[54]);
//hora_counter_int = hora_counter2.toInt();
hora_counter2 = String(hora_counter_int,DEC);
index_data = ano2+mes2+dia2+hora_counter2;
Serial.print("INDEX TESTE: ");
Serial.println(index_data);
Serial.print("Status: ");
Serial.println(status_esp);
Serial.print("Backup index: ");
Serial.println(array_backup_index);
Serial.print("Backup string: ");
for (int abc = 0; abc<=array_backup_index; abc++){
  Serial.println(arrayBackup[abc]);
}
if (status_esp == String(0)) {
  Serial1.end();
  ESP.restart();
}
}
//core1
void codeForTask1( void * parameter)
{
  for (;;) {

    delay(10000);
    readReset();

  }

}

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);

  //timer, callback, interrupção de borda
  Serial1.end();

  Serial.begin(115200);
  //Serial1.begin(110);
  WiFi.disconnect();
  //WiFi.begin("GVT-128B", "4807093708");

```

```
//WiFi.begin("TVC", "504b2014");  
WiFi.begin("AndroidAP", "qej3014");  
//wifiManager.autoConnect("EMSystem", "1234");
```

```
q1Backup[0] = 35;  
q2Backup[0] = 227;  
q3Backup[0] = 2;  
q4Backup[0] = 150;  
q5Backup[0] = 0;  
q6Backup[0] = 0;  
q7Backup[0] = 0;  
q8Backup[0] = 61;  
crcBackup[0] = 61;  
arrayBackup[0] = "2019011058";
```

```
q1Backup[1] = 35;  
q2Backup[1] = 227;  
q3Backup[1] = 2;  
q4Backup[1] = 250;  
q5Backup[1] = 0;  
q6Backup[1] = 0;  
q7Backup[1] = 0;  
q8Backup[1] = 61;  
crcBackup[1] = 61;  
arrayBackup[1] = "2019011079";  
// connectToWifi();
```

```
Serial.println(dia2);  
Serial.println(mes2);  
Serial.println(ano2);  
Serial.println(hora_counter2);  
index_data = ano2 + mes2 + dia2 + hora_counter2;  
Serial.println(index_data);
```

```
delay(2000);
```

```
//TESTES DE BACKUP  
alert(user_id, 2);
```

```
xTaskCreatePinnedToCore(  
  codeForTask1,  
  "Task_1",  
  8192,  
  NULL,  
  1,
```

```

    &Task1,
    0);

    ArduinoOTA.setHostname("myesp32");

    // Define a senha (opcional)
    ArduinoOTA.setPassword("123");

    // É possível definir uma criptografia hash md5 para a senha usando a função
    "setPasswordHash"
    // Exemplo de MD5 para senha "admin" = 21232f297a57a5a743894a0e4a801fc3
    // ArduinoOTA.setPasswordHash("21232f297a57a5a743894a0e4a801fc3"); q
    //define o que será executado quando o ArduinoOTA iniciar
    ArduinoOTA.onStart( startOTA ); //startOTA é uma função criada para simplificar o código

    //define o que será executado quando o ArduinoOTA terminar
    ArduinoOTA.onEnd( endOTA ); //endOTA é uma função criada para simplificar o código

    //define o que será executado quando o ArduinoOTA estiver gravando
    ArduinoOTA.onProgress( progressOTA ); //progressOTA é uma função criada para
    simplificar o código

    //define o que será executado quando o ArduinoOTA encontrar um erro
    ArduinoOTA.onError( errorOTA );//errorOTA é uma função criada para simplificar o código

    //inicializa ArduinoOTA
    ArduinoOTA.begin();

    timeFuture = millis() + interval;

    timeSend = millis() + 20000;

    nao_le = millis() + 60000;

    timer = timerBegin(0, 80, true); //timerID 0, div 80
    timerAttachInterrupt(timer, &resetModule, true);
    //timer, tempo (us), repetição
    timerAlarmWrite(timer, 30000000, true);
    timerAlarmEnable(timer); //habilita a interrupção

    timeFuture = millis() + 2000;
    Serial1.begin(110);
}
void loop() {

    // put your main code here, to run repeatedly:
    ArduinoOTA.handle();
    timerWrite(timer, 0);

```

```

// hora_counter_int++;
// index_data = ano2+mes2+dia2+hora_counter_int;
// Serial.println(index_data);

//if (millis() < timeFuture) {
if (Serial1.available() > 0) {
  frameMedidor[i] = Serial1.read();
  Serial.print(i);
  Serial.print(" | ");
  Serial.println(frameMedidor[i], DEC);
  //Serial.print(" || ");
  i++;
  timeFuture = millis() + interval;
  tratamento = false;
  flag_nao_envia = false;
  timeFuture = millis() + interval;
  nao_le = millis() + 60000;
  // }
} else if ((tratamento == false) && (i == 8)) {

  Serial.print("nao_envia_test: ");
  Serial.println(nao_envia_test);
  flag_nao_envia = false;
  tratamento = true;
  i = 0;

  for (n = 0; n < 8; n++) {
    frameBackupMedidor[n] = frameMedidor[n];
  }

  lastTime = temporestante;
  temporestantelsb = frameBackupMedidor[0];
  temporestantemsb = frameBackupMedidor[1] << 8;
  temporestantemsb = temporestantemsb & 4095;
  temporestante = temporestantemsb + temporestantelsb;
  Serial.println("");;
  Serial.println("");;
  Serial.println(temporestante);
  Serial.println("");;
  Serial.println("");;
  flag_enviando = true;
  for (n = 0; n <= 7; n++) {
    Serial.println(frameBackupMedidor[n], DEC);
  }

  Serial.println("");

  for (n = 0; n < 7; n++) {

```

```

if (n == 0)
{
  all_xor = frameBackupMedidor[n];
} else {
  all_xor = all_xor ^ frameBackupMedidor[n];
}
}
all_xor = ~all_xor;
Serial.print("CHECKSUM CALCULADO: ");
Serial.println(all_xor, DEC);
if (all_xor == frameBackupMedidor[7]) {
  Serial.println("true");
  check_sum_test = 0;
} else
{
  Serial.println("false");
  check_sum_test++;
  if (check_sum_test == 5) {
    Serial1.end();
    check_sum_test = 0;
    alert(user_id, 3);

    Serial.println("ALERTA CHECKSUM");
    delayMicroseconds(9090);
    Serial1.begin(110);
    //ESP.restart();
  }
}

if (temporestante == 0 && temporestante != lastTime) {
  //array_index[] = index_data = ano2+mes2+dia2+hora_counter_int;
  hora_counter_int++;
  if (hora_counter_int == 96) {
    hora_counter_int = 0;
  }
  if (status_esp != String(2) || enviando_backup == false) {
    sendData(true);
  }
} else if (millis() > timeSend && temporestante > 60) {
  if (status_esp != String(2) || enviando_backup == false) {
    sendData(false);
  }
}

// Serial.println("");
// nao_envia = millis()+60000;
} else if (millis() > nao_le) {
  Serial.println("ALERTA NAO LE");
  alert(user_id, 4);
}

```

```

    Serial1.end();
    ESP.restart();
} else {
    i = 0;
    timeFuture = millis() + interval;
}
if (nao_envia_test == 4) {
    nao_envia_test = 0;
    Serial.println("ALERTA NAO ENVIA");
    //alert(user_id, 5);
    //Serial1.end();
    //ESP.restart();
    //connectToWifi();
}

}

char calc_checksum(void) {
    for (n = 0; n < 7; n++) {
        if (n == 0)
        {
            all_xor = frameBackupMedidor[n];
        } else {
            all_xor = all_xor ^ frameBackupMedidor[n];
        }
    }
    all_xor = ~all_xor;
    return all_xor;
}

char calc_new_checksum(void) {
    for (n = 0; n < 7; n++) {
        if (n == 0)
        {
            all_xor = correctFrameMedidor[n];
        } else {
            all_xor = all_xor ^ correctFrameMedidor[n];
        }
    }
    all_xor = ~all_xor;
    return all_xor;
}

void sendData(bool incrementar) {

```

```

stringToBeSent = "";
for (n = 0; n < 8; n++) {
    frameBackupMedidor[n] = frameMedidor[n];
}
for (n = 0; n < 8; n++) {
    //if (n == 3) {
    //    stringToBeSent = stringToBeSent + "&q" + String(n + 1) + "=" + String(400);
    //    var_send_test++;
    // } else {
    stringToBeSent = stringToBeSent + "&q" + String(n + 1) + "=" +
String(frameBackupMedidor[n], DEC);
    //}
}
//colocar checksum
checksum = calc_checksum();
stringToBeSent = stringToBeSent + "&crc=" + String(checksum, DEC);

// if (incrementar == true) {
//    stringToBeSent = stringToBeSent + "&incrementar=true";
// }
Serial.println(stringOne + stringToBeSent);
http.begin(stringOne + stringToBeSent);
int httpCode = http.GET();
if (incrementar == true && httpCode != 200) {
    salvarBackup();
}
if (httpCode > 0) {
    String payload = http.getString();
    Serial.println(httpCode);
    Serial.println(payload);
} else {
    Serial.println("Error on HTTP request");
    nao_envia_test++;
}

if (httpCode != 200) {
    nao_envia_test++;
} else if (httpCode == 200) {
    nao_envia_test = 0;
}

http.end();
//ESP.restart();
if (start_enviado == false) {
    timeSendAlert = millis() + 10000;
}

```



```

timeSend = millis() + 120000;

}

void alert(int user_id, int type) {
  //int type define o tipo de alerta que devo receber
  stringToBeSent = alertString + String(user_id) + "&type=" + String(type);
  Serial.println("");
  Serial.println(stringToBeSent);
  Serial.println("");
  http.begin(stringToBeSent);
  int httpCode = http.GET();
  if (httpCode > 0) {
    String payload = http.getString();
    Serial.println(httpCode);
    Serial.println(payload);
    if (type == 2 && httpCode == 200) {
      start_enviado = true;
    }
  } else {
    Serial.println("Error on HTTP request");
  }
}

int attempt_to_correct(void) {
  //tentar corrigir o check sum false, mas acho que é mais fácil só mudar o index

  int sum_index = 1;
  int new_index;
  for (n = 0; n < 8; n++) {
    new_index = n + sum_index;
    if (new_index >= 8) {
      new_index = n + sum_index - 8;
    }
    correctFrameMedidor[new_index] = frameMedidor[n];
  }
  int new_crc = calc_new_checksum();

  if (new_crc == correctFrameMedidor[8]) {
    return sum_index;
  } else {
    sum_index++;
    if (sum_index == 8) {
      return 0;
    }
  }
}

```

```

void resetModule() {
  //ets_printf("(watchdog) reiniciar\n"); //imprime no log
  //esp_restart_noos(); //reinicia o chip
  Serial1.end();
  ESP.restart();
}

//funções de exibição dos estágios de upload (start, progress, end e error) do ArduinoOTA
void startOTA()
{
  String type;

  //caso a atualização esteja sendo gravada na memória flash externa, então informa "flash"
  if (ArduinoOTA.getCommand() == U_FLASH)
    type = "flash";
  else //caso a atualização seja feita pela memória interna (file system), então informa
  "filesystem"
    type = "filesystem"; // U_SPIFFS

  //exibe mensagem junto ao tipo de gravação
  Serial.println("Start updating " + type);
}

//exibe mensagem
void endOTA()
{
  Serial.println("\nEnd");
}

//exibe progresso em porcentagem
void progressOTA(unsigned int progress, unsigned int total)
{
  Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
}

//caso aconteça algum erro, exibe especificamente o tipo do erro
void errorOTA(ota_error_t error)
{
  Serial.printf("Error[%u]: ", error);

  if (error == OTA_AUTH_ERROR)
    Serial.println("Auth Failed");
  else if (error == OTA_BEGIN_ERROR)
    Serial.println("Begin Failed");
  else if (error == OTA_CONNECT_ERROR)
    Serial.println("Connect Failed");
}

```

```

else if (error == OTA_RECEIVE_ERROR)
  Serial.println("Receive Failed");
else if (error == OTA_END_ERROR)
  Serial.println("End Failed");
}

void connectToWifi(void) {
  while (WiFi.status() != WL_CONNECTED) {
    WiFi.begin("TVC", "504b2014");
    //WiFi.begin("AndroidAP", "qeju3014");
    // WiFi.begin("GVT-128B", "4807093708");
    delay(20000);
    Serial.println("Connecting to WiFi..");
    Serial.println(WiFi.status());

    // connection_attempts++;
    // if (connection_attempts == 6) {
    //   Serial1.end();
    //   ESP.restart();
    // }
  }
  Serial.println("Connected to the WiFi network");
  Serial.println(WiFi.localIP());

  http.begin("http://104.131.45.58/dados_iniciais.json");
  int httpCode2 = http.GET();
  if (httpCode2 > 0) {
    payload2 = http.getString();
    Serial.println(httpCode2);
    Serial.println(payload2);
  } else {
    Serial.println("Error on HTTP request");
  }

  dia2 = String(payload2[8]) + String(payload2[9]);
  mes2 = String(payload2[19]) + String(payload2[20]);
  ano2 = String(payload2[29]) + String(payload2[30]) + String(payload2[31]) +
String(payload2[32]);
  hora_counter2 = String(payload2[42]) + String(payload2[43]);
  hora_counter_int = hora_counter2.toInt();
  enviarBackup();
}

void salvarBackup(void) {
  hora_counter2 = String(hora_counter_int, DEC);
  arrayBackup[array_backup_index] = ano2 + mes2 + dia2 + hora_counter2;
  q1Backup[array_backup_index] = frameBackupMedidor[0];
}

```

```

q2Backup[array_backup_index] = frameBackupMedidor[1];
q3Backup[array_backup_index] = frameBackupMedidor[2];
q4Backup[array_backup_index] = frameBackupMedidor[3];
q5Backup[array_backup_index] = frameBackupMedidor[4];
q6Backup[array_backup_index] = frameBackupMedidor[5];
q7Backup[array_backup_index] = frameBackupMedidor[6];
q8Backup[array_backup_index] = frameBackupMedidor[7];
crcBackup[array_backup_index] = checksum;
//arrayBackup[array_backup_index] =
// hora_counter_int++;
array_backup_index++;

}

void enviarBackup(void) {
    enviando_backup = true;
    int o = -1;
    for (int p = 0; p < 2880; p++) {
        if (arrayBackup[p] == "0") {
            break;
        }
        o++;
    }
    if (o != -1) {
        for (int p = 0; p <= o; p++) {
            sendBackupData(p);
        }
    }

    array_backup_index = 0;
    enviando_backup = false;
}

int lookForIndex(String searchedString) {
    int wantedpos = -1;
    for (int i = 0; i < 2880; i++) {
        if (searchedString == arrayBackup[i]) {
            wantedpos = i;
        }
    }
    return wantedpos;
}

void sendBackupData(int index) {
    backupData[0] = q1Backup[index];
    backupData[1] = q2Backup[index];
    backupData[2] = q3Backup[index];
    backupData[3] = q4Backup[index];
    backupData[4] = q5Backup[index];
}

```

```

backupData[5] = q6Backup[index];
backupData[6] = q7Backup[index];
backupData[7] = q8Backup[index];
backupData[8] = crcBackup[index];
//backupData[9] = arrayBackup[index];

```

```
String backupStringHttp;
```

```

backupStringHttp = "http://104.131.45.58/backup.json?user_id=1&q1=";
backupStringHttp = backupStringHttp + String(backupData[0], DEC);
for (n = 1; n < 8; n++) {
    backupStringHttp = backupStringHttp + "&q" + String(n + 1) + "=" +
String(backupData[n], DEC);
}

```

```

backupStringHttp = backupStringHttp + "&crc=" + String(backupData[8], DEC);
backupStringHttp = backupStringHttp + "&index=" + String(arrayBackup[index]);
Serial.println(backupStringHttp);
http.begin(backupStringHttp);

```

```

int httpCode2 = http.GET();
if (httpCode2 > 0) {
    payload2 = http.getString();
    Serial.println(httpCode2);
    Serial.println(payload2);
    arrayBackup[index] = "0";
} else {
    Serial.println("Error on HTTP request");
}
}

```

Apêndice C – Artigo ISIE