

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Eduarda Coelho Becker

METODOLOGIA DE BALANÇO DE ENERGIA ELÉTRICA PARA
DETECTAR PERDAS NÃO TÉCNICAS EM SISTEMAS DE
DISTRIBUIÇÃO

Santa Maria, RS
2023

Eduarda Coelho Becker

**METODOLOGIA DE BALANÇO DE ENERGIA ELÉTRICA PARA DETECTAR
PERDAS NÃO TÉCNICAS EM SISTEMAS DE DISTRIBUIÇÃO**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica, da Universidade Federal de Santa Maria (UFSM, RS), como requisito para obtenção do título de Engenheira Eletricista.

Orientador: Prof. Daniel Pinheiro Bernardon

Santa Maria, RS
2023

Eduarda Coelho Becker

**METODOLOGIA DE BALANÇO DE ENERGIA ELÉTRICA PARA DETECTAR
PERDAS NÃO TÉCNICAS EM SISTEMAS DE DISTRIBUIÇÃO**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica, da Universidade Federal de Santa Maria (UFSM, RS), como requisito para obtenção do título de **Engenheira Eletricista**.

Aprovado em 02 de junho de 2023:

Daniel Pinheiro Bernardon, Dr. (DESP-UFSM) (videoconferência)
(Orientador)

Marina Camponogara, Me. (UFSM) (videoconferência)

Mauro dos Santos Ortiz, Me. (UFSM) (videoconferência)

Santa Maria, RS
2023

RESUMO

METODOLOGIA DE BALANÇO DE ENERGIA ELÉTRICA PARA DETECTAR PERDAS NÃO TÉCNICAS EM SISTEMAS DE DISTRIBUIÇÃO

AUTORA: Eduarda Coelho Becker

ORIENTADOR: Daniel Pinheiro Bernardon

O presente trabalho foi produzido concomitantemente ao projeto de extensão intitulado "Sistema para Detectar Perdas Não Técnicas em Áreas de Irrigação Empregando Técnicas de Inteligência Artificial" e tem como objetivo determinar as perdas não técnicas de energia elétrica em sistemas de distribuição utilizando a metodologia do balanço energético. As perdas de energia são quantificadas por tipo, podendo ser perdas técnicas ou não técnicas (comerciais) e ambas em conjunto representam as perdas globais do sistema. As perdas globais podem ser calculadas a partir da diferença entre a energia fornecida pela concessionária e a energia demandada pelas unidades consumidoras (UCs). As perdas técnicas são inerentes aos processos de geração, transporte e conversão de energia ao longo do sistema, tendo o Efeito Joule como um dos exemplos mais conhecidos. Por outro lado, as perdas não técnicas referem-se à energia que foi consumida, mas que não foi faturada pela empresa fornecedora. Sendo assim, para estimar o valor das perdas não técnicas no sistema de distribuição, a partir do balanço energético, foram utilizados dados reais de consumo e de fornecimento, bem como informações das redes elétricas, disponibilizados pelo Grupo CPFL Energia. A linguagem Python foi utilizada na automatização de uma ferramenta capaz de realizar o balanço energético a partir dos dados de entrada disponibilizados pela concessionária. A metodologia empregada fornece como dados de saída as perdas globais, técnicas e não técnicas dos sistemas de distribuição analisados e indica os alimentadores com possíveis UCs irregulares. Uma vez identificados a região ou alimentadores passíveis de inconformidades, é facilitada à empresa de energia a tomada de ações para minimizar as perdas não técnicas nas redes elétricas de distribuição.

Palavras-chave: Balanço de energia. Perdas elétricas. Python. OpenDSS.

ABSTRACT

ELECTRIC ENERGY BALANCE METHODOLOGY TO DETECT NON-TECHNICAL LOSSES IN DISTRIBUTION SYSTEMS.

AUTHOR: Eduarda Coelho Becker
ADVISOR: Daniel Pinheiro Bernardon

The present work was produced concurrently with the extension project entitled "System for Detecting Non-Technical Losses in Irrigation Areas Using Artificial Intelligence Techniques" and aims to determine non-technical losses of electrical energy in distribution systems using the energy balance methodology. Energy losses are quantified by type, which can be technical or non-technical (commercial) losses, and both together represent the overall losses of the system. Overall losses can be calculated based on the difference between the energy supplied by the utility company and the energy demanded by consumer units (UCs). Technical losses are inherent in the processes of generation, transmission, and energy conversion throughout the system, with Joule's Effect being one of the most well-known examples. On the other hand, non-technical losses refer to energy that was consumed but not billed by the supplying company. Therefore, to estimate the value of non-technical losses in the distribution system, based on the energy balance, real consumption and supply data, as well as information on the electrical networks provided by the CPFL Energia Group, were used. The Python language was used to automate a tool capable of performing the energy balance using the input data provided by the utility company. The employed methodology provides as output data the overall, technical, and non-technical losses of the analyzed distribution systems and identifies feeders with possible irregular consumer units. Once regions or feeders prone to non-conformities are identified, it facilitates the energy company in taking actions to minimize non-technical losses in the distribution electrical networks.

Keywords: Energy balance. Electrical losses. Python. OpenDSS.

LISTA DE FIGURAS

Figura 1 - Balanço de Energia	18
Figura 2 - Curva diária de potência ativa para o alimentador A em 04 de maio de 2020, antes e após a compensação de medições zeradas.	26
Figura 3 - Fluxograma do algoritmo desenvolvido no Python.....	30
Figura 4 - Topologia da rede elétrica do Alimentador 1 de distribuição.....	34
Figura 5 - Topologia da rede elétrica do Alimentador 2 de distribuição.....	34
Figura 6 - Resultado de Perda Técnica no período de 24h para o Alimentador 1.....	36
Figura 7 - Potência Ativa Fornecida pelo Alimentador 1 na SE no período de 24h.	37
Figura 8 - Resultado de Perda Técnica no período de 24h para o Alimentador 2.....	38
Figura 9 - Potência Ativa Fornecida pelo Alimentador 2 na SE no período de 24h.	39
Figura 10 - Percentual de PG, PT e PNT da energia fornecida pelo Alimentador 1 de distribuição.....	40
Figura 11 - Percentual de PG, PT e PNT da energia fornecida pelo Alimentador 2 de distribuição.....	40

LISTA DE TABELAS

Tabela 1 - Compensação de dados de consumo - Grupo A.....	22
Tabela 2 - Compensação de dados de consumo - Grupo B.....	23
Tabela 3 - Consumo total 2020/01	24
Tabela 4 - Energia fornecida total – 2020/01	27
Tabela 5 - Dados dos alimentadores utilizados.	33
Tabela 6 - Resultado da Perda Global para os Alimentadores 1 e 2.	35
Tabela 7 - Resultado de Perda Técnica mensal para os Alimentadores 1 e 2	39
Tabela 8 - Resultado de Perda Não Técnica para os Alimentadores 1 e 2.....	40

LISTA DE ABREVIATURAS E SIGLAS

ANEEL	Agência Nacional de Energia Elétrica
CA	Corrente Alternada
CC	Corrente Contínua
GIS	<i>Geographic Information System</i>
HVDC	<i>High Voltage Direct Current</i>
IA	Inteligência Artificial
OpenDSS	<i>Open Distribution System Simulator</i>
PG	Perda Global
PNT	Perda Não Técnica
PT	Perda Técnica
SCADA	<i>Supervisory Control And Data Acquisition</i>
SEP	Sistema Elétrico de Potência
UCs	Unidades Consumidoras
VVC	<i>Volt/Var Control</i>
WLS-SE	<i>Weighted Least Squares State Estimator</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVO GERAL.....	11
1.2	OBJETIVOS ESPECÍFICOS	11
1.3	ORGANIZAÇÃO DOS CAPÍTULOS.....	11
2	REFERENCIAL TEÓRICO	13
2.1	SISTEMA ELÉTRICO DE POTÊNCIA	13
2.1.1	Geração.....	13
2.1.2	Transmissão	13
2.1.3	Distribuição	14
2.2	PERDAS ELÉTRICAS	14
2.2.1	Perdas Técnicas de Energia Elétrica	15
2.2.2	Perdas Não Técnicas de Energia Elétrica	15
2.3	BALANÇO DE ENERGIA.....	16
3	METODOLOGIA DE BALANÇO ENERGÉTICO PARA SISTEMAS DE DISTRIBUIÇÃO	18
3.1	DESCRIÇÃO DO MÉTODO PROPOSTO	18
3.2	DADOS UTILIZADOS	20
3.2.1	Dados de Faturamento.....	20
3.2.1.1	Tratamento de dados	20
3.2.2	Medições SCADA	25
3.2.2.1	Tratamento de dados	26
3.2.3	Redes Elétricas.....	27
3.3	IMPLEMENTAÇÃO DA METODOLOGIA	28
3.3.1	OpenDSS	31
3.3.2	Python.....	32
4	RESULTADOS E DISCUSSÕES	33
4.1	RESULTADOS DO CÁLCULO DE PERDA GLOBAL.....	35
4.2	RESULTADO DA ESTIMATIVA DAS PERDAS TÉCNICAS.	35
4.3	RESULTADO DAS PERDAS NÃO TÉCNICAS.....	39

5	CONCLUSÃO	42
	REFERÊNCIAS BIBLIOGRÁFICAS	43
	APÊNDICE A	45

1 INTRODUÇÃO

A crescente demanda por energia elétrica torna cada vez mais necessária a otimização das redes elétricas de distribuição. Com o advento de novas tecnologias e a eventual popularização do carro elétrico, por exemplo, a expansão das redes elétricas ou até mesmo a troca de equipamentos existentes pode ser iminente. Entretanto, é possível que a otimização do sistema postergue ou evite investimentos em infraestrutura, aliviando o caixa das empresas de energia e diminuindo o custo repassado ao consumidor na forma de tarifa. No segmento de distribuição, por exemplo, uma forma de otimização é a redução das perdas de energia elétrica.

Atualmente, muitos estudos estão sendo desenvolvidos com o objetivo de, a partir de cálculos e estimativas, determinar o montante de perdas e identificar onde elas ocorrem, pois este é um desafio para as concessionárias que buscam diminuir as perdas de energia nas redes elétricas. Segundo Bernardon *et al.* (2007), o conhecimento das perdas de energia elétrica de forma geral, ou seja, perdas globais (PG) do sistema, não é suficiente para contribuir com a tomada de decisões para redução de perdas no segmento de distribuição. Segundo os autores, os fatores que determinam a ação a ser desenvolvida e quais os recursos necessários são o tipo de perda e seu local de ocorrência.

De acordo com Bretas *et al.* (2020), as PGs nos sistemas de potência são definidas como a diferença entre a energia elétrica injetada pelo alimentador e a energia faturada pelas unidades consumidoras (UCs). As perdas de energia podem ser classificadas em dois tipos: técnicas e não técnicas. Perdas técnicas (PTs) são inerentes ao processo de transporte de eletricidade, como perdas por aquecimento (efeito Joule) e perdas no núcleo e nos enrolamentos de transformadores. Já as perdas não técnicas (PNTs) são referentes à parcela da energia que foi entregue, mas que não foi faturada pela concessionária, podendo ocorrer devido a vários fatores, como erros de medição, fraudes ou roubos.

As PGs de energia elétrica podem ser estimadas com precisão através de métodos de balanço energético e fluxo de potência, incluindo valores de PTs e PNTs (BIAZZI *et al.*, 2021). Conforme Madrigal, Rico e Uzcategui (2017) as PTs podem ser calculadas a partir de diversos métodos dependendo dos dados disponíveis do sistema, como fluxo de potência da rede elétrica, cálculos estatísticos, entre outros. Já as PNTs, podem ser obtidas pela diferença entre as PGs do sistema e as PTs. Desta forma, com uma estimativa precisa das PTs é possível obter um bom resultado para as PNTs.

1.1 OBJETIVO GERAL

Desenvolvimento e aplicação de um método de balanço energético para estimativa de PNTs em alimentadores da rede de distribuição de energia elétrica.

1.2 OBJETIVOS ESPECÍFICOS

Este trabalho possui os seguintes objetivos específicos:

- Desenvolver um algoritmo de balanço energético em Python para estimar as perdas elétricas das redes de distribuição, utilizando a interface entre Python e Open Distribution System Simulator (OpenDSS).
- Utilizar dados reais de consumo, fornecimento e de redes elétricas disponibilizados pelo Grupo CPFL Energia como entrada do algoritmo de balanço de energia. Os dados de fornecimento são obtidos a partir do sistema SCADA (do inglês, Supervisory Control And Data Acquisition) e os dados de rede elétrica são extraídos do sistema de informações geográficas (GIS, do inglês Geographic Information System) para o formato utilizado no OpenDSS.
- Estimar o valor de PTs nos alimentadores através de simulações de fluxo de potência no OpenDSS. O fluxo de potência é realizado para o período de interesse e com o resultado pode-se avaliar de quanto foi a perda técnica envolvida no processo de distribuição.
- Determinar quais alimentadores apresentam potenciais chances de ter consumidores ilegais conectados à rede, com base nos resultados das PNTs.

1.3 ORGANIZAÇÃO DOS CAPÍTULOS

Este trabalho foi organizado da seguinte forma:

- No primeiro capítulo, é introduzido o tema proposto junto às suas considerações iniciais, bem como seus objetivos;
- No segundo capítulo, é fornecida a contextualização do sistema elétrico de potência, com uma base teórica sobre as perdas elétricas encontradas no sistema de distribuição e o conceito de balanço energético;

- No terceiro capítulo, é apresentada a metodologia do trabalho, incluindo os dados de entrada e o procedimento realizado para exemplificar o processo automático em Python;
- No quarto capítulo, são apresentados os resultados para dois estudos de casos e discussões dos dados obtidos;
- Por fim, o quinto capítulo apresenta as principais conclusões e considerações finais deste trabalho.

2 REFERENCIAL TEÓRICO

Nas seções seguintes, será discutida a fundamentação teórica utilizada como base para o desenvolvimento deste trabalho.

2.1 SISTEMA ELÉTRICO DE POTÊNCIA

O sistema elétrico de potência (SEP) é dividido em três principais componentes: geração, transmissão e distribuição. A energia elétrica é produzida nas usinas geradoras, sendo elevada a tensão nas subestações elevadoras e transportada através de linhas de transmissão até as subestações rebaixadoras, usualmente mais próximas dos grandes centros urbanos ou industriais. Por fim, o segmento de distribuição é responsável pela entrega da energia ao consumidor final, que podem ser grandes consumidores (média tensão) ou consumidores regulares (baixa tensão) (SILVA, 2018).

2.1.1 Geração

A geração de energia elétrica ocorre em usinas geradoras que utilizam diferentes fontes, tais como hidrelétricas, biomassa, gás natural, fotovoltaica, nuclear, entre outras. Atualmente, a geração de energia elétrica no Brasil tem apresentado um cenário bastante otimista para as fontes de energia renováveis, representando 83,56% da matriz elétrica brasileira, no ano de 2023. Desses, a energia hídrica é responsável por 57,11%, a energia eólica por 13,33%, a biomassa por 8,63% e a energia solar por 4,49%, de acordo com os dados disponibilizados pela Agência Nacional de Energia Elétrica (ANEEL).

2.1.2 Transmissão

O sistema de transmissão realiza o transporte da energia gerada nas usinas até os centros de consumo, onde a tensão é rebaixada para os níveis de distribuição. Segundo Silveira *et al.* (2022), as linhas de transmissão podem ser constituídas de diversos tipos de torres de transmissão, de acordo com o nível de tensão da linha. Essas torres são responsáveis por dar suporte aos condutores de eletricidade, suspensos através de isoladores (que evitam o contato da torre com os condutores).

Ainda de acordo com Silveira *et al.* (2022), a transmissão pode ser feita em corrente alternada (CA) ou corrente contínua (CC). A transmissão CC, embora menos utilizada, é feita em níveis de extra alta tensão (HVDC, do inglês High Voltage Direct Current) e é muito útil em longas distâncias. Os autores mencionam os baixos níveis de perda como uma grande vantagem da transmissão em CC em relação à transmissão CA, embora os seus altos custos de operação e das estações conversoras sejam pontos negativos.

2.1.3 Distribuição

A rede de distribuição de energia elétrica é a parcela do SEP encarregada de conduzir a energia, desde a subestação rebaixadora na etapa de transmissão, até o consumidor final (SILVEIRA *et al.*, 2022). Esse sistema consiste em uma rede primária e outra secundária, cuja tensão de linha para o sul do Brasil é, geralmente, 13,8 kV e 0,38 kV, respectivamente. Dentre os principais elementos de uma rede de distribuição podem-se citar os alimentadores primários das subestações de distribuição (redes aéreas ou subterrâneas), transformadores, equipamentos de medição, proteção e outros componentes das redes elétricas (SILVA, 2018).

2.2 PERDAS ELÉTRICAS

De acordo com Aguëro (2012), o conhecimento das perdas de energia é fundamental para a otimização do sistema de distribuição de energia, uma vez que permite a realização de análises, planejamentos e operações de redução de perdas, contribuindo para o aumento da eficiência do sistema elétrico. Segundo Biazzini *et al.* (2021) as perdas de energia constituem um grande desafio na operação de sistemas de distribuição, pois afetam a qualidade de energia, faturamento das concessionárias e, conseqüentemente, aumento das tarifas de eletricidade para os demais consumidores.

Ainda de acordo com Biazzini *et al.* (2021), as PGs do sistema podem ser obtidas pela diferença entre a energia fornecida pelos alimentadores das subestações e a energia faturada pela empresa distribuidora de energia. As perdas ocorrem durante todo o processo realizado para levar a energia até o consumidor, incluindo a utilização pelo cliente. Tais perdas podem ser dos tipos, PT e PNT, e sua minimização evita futuros empreendimentos e expansão desnecessários das redes. Além disso, o retorno financeiro sobre o investimento inicial da rede é otimizado (DAVIDSON, 2003).

De acordo com Davidson (2003), com a estimativa das perdas elétricas, a empresa de fornecimento de energia pode prever os custos operacionais futuros para manter o fornecimento de energia às UCs. Também é possível estimar a vida útil da rede elétrica e seus investimentos essenciais com manutenção durante esse período.

2.2.1 Perdas Técnicas de Energia Elétrica

No sistema de distribuição, o processo realizado para levar a energia elétrica até o consumidor final é suscetível a PT devido a vários fatores, como o Efeito Joule (associado ao aquecimento) nos condutores da rede e perdas no núcleo e nos enrolamentos dos transformadores. Esse tipo de perda pode ser reduzido com a troca de condutores da rede, manutenção de equipamentos e substituição de transformadores por outros de maior capacidade e/ou melhor eficiência (GARCIA *et al.*, 2021).

Além disso, de acordo com Biazzini *et al.* (2021), técnicas de otimização para operação dos equipamentos em campo podem ser utilizadas para reduzir as PTs de energia, como por exemplo, o controle heurístico de tensão/reativo (VVC, do inglês, Volt/Var Control) que busca a otimização do sistema a partir de análises combinatórias reduzidas.

De acordo com Bernardon *et al.* (2007), as PTs no sistema de distribuição vão além das perdas em condutores e transformadores, porém são desprezadas por serem muito pequenas. Essas perdas podem ocorrer em ramais de ligação, medidores, reguladores de tensão, equipamentos de proteção e operação de manobras, entre outros.

Conforme Agüero (2012), as PTs podem ser estimadas a partir de métodos e suposições aproximados. Por exemplo, uma vez adquirido o valor de PT média no tempo, ou seja, em horas, é possível determinar as perdas mensais e/ou anuais multiplicando o valor obtido pelo tempo.

2.2.2 Perdas Não Técnicas de Energia Elétrica

De acordo com Trevizan, Bretas e Rossini (2018), as PNTs representam a parcela das PGs que foi entregue ao consumidor, mas que não foi faturada pela concessionária de energia. As PNTs prejudicam a eficiência energética e a lucratividade das distribuidoras de energia que repassam os custos aos consumidores finais, resultando em um aumento na conta de energia. Conforme Madrigal, Rico e Uzcategui (2017), as PNTs podem ser ocasionadas devido a vários

fatores, por exemplo, roubo¹ de energia por usuários externos (que não são clientes da empresa), fraudes² de energia por clientes, falhas de medição e erros de faturamento.

Além disso, segundo Davidson (2003), as PNTs podem ser originadas devido à existência de medidores e equipamentos defeituosos assim como medição inadequada, leitura de medidores imprecisas, perda ou danos a equipamentos de proteção, erro intrínseco de medidores, cabos e/ou equipamentos de comutação e fornecimentos não computados, como iluminação pública e consumo agrícola. Além disso, as PNTs são mais improváveis de serem encontradas em sistemas de geração e transmissão de energia elétrica, sua ocorrência é mais comum no segmento de distribuição já que estes estão mais suscetíveis a serem afetados por práticas ilícitas (MADRIGAL; RICO; UZCATEGUI, 2017).

Conforme Silva (2018), as concessionárias de energia deslocam equipes de fiscalização até as UCs com o objetivo de reduzir as PNTs nas redes de distribuição. No entanto, essa abordagem não é eficaz, uma vez que a concessionária não possui informações precisas sobre possíveis irregularidades. Por essa razão, o estudo e a estimativa das PNTs e de sua localização são tão importantes. Dessa forma, as concessionárias evitam gastos com deslocamentos e desperdício de tempo das equipes.

2.3 BALANÇO DE ENERGIA

Na maior parte dos estudos de perdas no sistema de distribuição, as PGs de energia elétrica são estimadas através de métodos de balanço energético, que consistem em subtrair a energia consumida pelos clientes da energia fornecida pelos alimentadores de distribuição. As PGs incluem PT e PNT e, por isso, é necessário realizar análises separadas para obter seus valores individualmente (AGÜERO, 2012).

Em Madrigal, Rico e Uzcategui (2017), uma metodologia para estimar as PNTs foi desenvolvida a partir de modelos matemáticos de regressão linear múltipla com intuito de obter um valor preciso de PNT possibilitando ações de controle e diminuição destas. Rossini *et al.* (2017) utilizou como método o estimador de estados de mínimos quadrados ponderados (WLS-SE, do inglês, Weighted Least Squares State Estimator) para identificar PT e PNT em todas as

¹ O furto de energia refere-se à ação de obter eletricidade de forma clandestina, sem passar pelos medidores ou sistemas de medição adequados.

² A fraude de energia ocorre quando há uma manipulação intencional das informações relacionadas ao consumo de energia para obter vantagens indevidas.

barras na rede de distribuição. Ainda, foi utilizado um modelo de estimativa de estados linear para analisar a relação entre a estimativa das perdas e os erros de medição.

Além disso, algoritmos genéticos também são utilizados para estimar perdas elétricas, pois realizam a reconfiguração do sistema para obter a otimização das PTs nas redes de distribuição de energia elétrica. Com o valor das PT, as PNT são estimadas por meio de métodos analíticos estatísticos, como realizado em Kassem, Badr e Ahmed (2013).

Em Leite e Mantovani (2018) a detecção de PNT foi realizada utilizando um gráfico de controle multivariado que determina uma região confiável para monitorar a grandeza aferida e um método de busca de caminho através do algoritmo “A-Star” para localizar onde a PNT ocorre. Esse estudo leva em consideração possíveis fraudes de energia na alteração de dados de consumo (perdas não técnicas) ocasionadas por ataques cibernéticos, devido às tecnologias da informação e comunicação recentemente adicionadas nos sistemas de distribuição, como medidores inteligentes.

Bernardon *et al.* (2007) desenvolveram um algoritmo capaz de determinar as PTs de potência ativa e reativa a partir do cálculo do fluxo de potência das redes elétricas. Para obter as PNT de potência ativa, foi utilizado o método do balanço de energia, que consiste em subtrair as PT das PG.

Para Bretas *et al.* (2020), a estimativa de PNT apenas com modelos de dados não é eficaz, uma vez que as PNT estão em constante evolução. Desta forma, a metodologia utilizada para estimar as PNT consiste em um *framework* híbrido composto pela combinação de modelos de dados e modelos de física para a solução do problema.

Sendo assim, pode-se perceber que existem várias técnicas presentes na literatura para estimar PNT. Em geral, a aplicabilidade de cada técnica depende dos dados disponíveis do sistema de distribuição. Neste trabalho, utilizou-se o software OpenDSS para estimar as PT em sistemas de distribuição a partir do cálculo do fluxo de potência. Posteriormente, para obter o valor das PNT, utilizou-se a metodologia do balanço de energia.

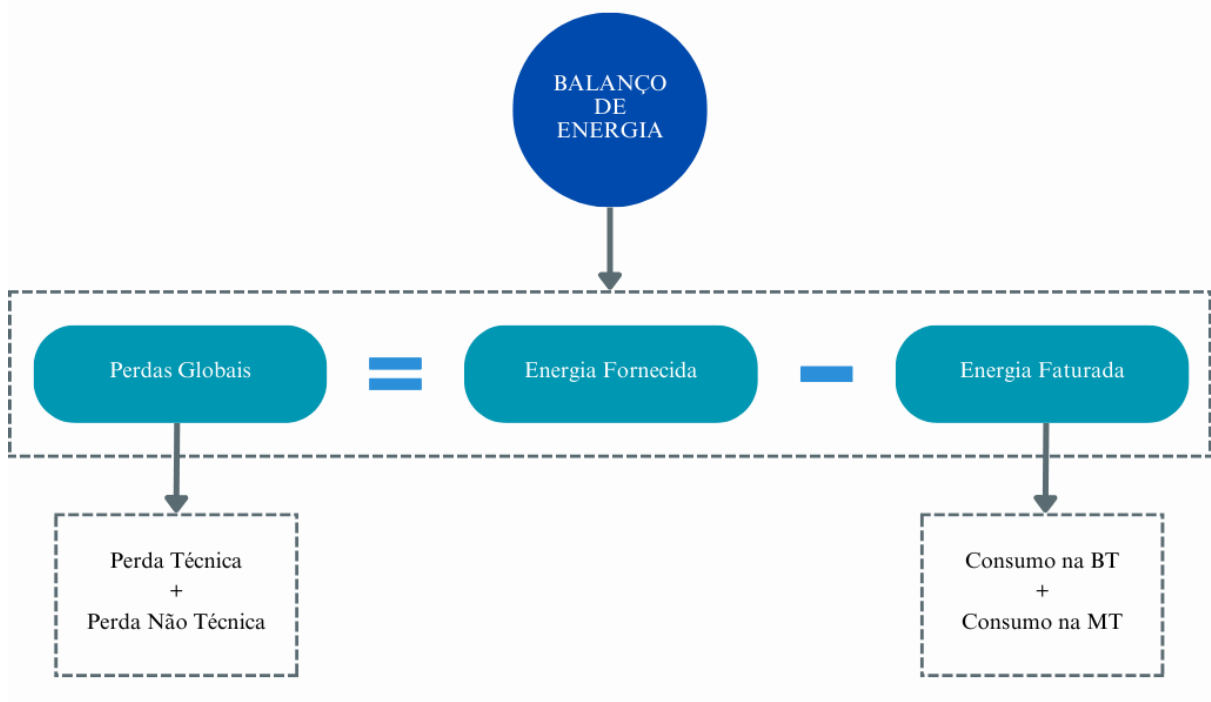
3 METODOLOGIA DE BALANÇO ENERGÉTICO PARA SISTEMAS DE DISTRIBUIÇÃO

Nessa seção, serão abordados tópicos referentes à metodologia utilizada para o desenvolvimento do trabalho, informações necessárias e aplicações práticas. Cada tópico abordará uma parte da metodologia do balanço energético, apresentando as equações necessárias para a realização dos cálculos e os ajustes necessários nos dados, para facilitar a compreensão do algoritmo desenvolvido. Por fim, na seção dedicada à implementação prática, será discutido como ocorreu o desenvolvimento do algoritmo e quais ferramentas foram necessárias para a sua criação.

3.1 DESCRIÇÃO DO MÉTODO PROPOSTO

O método do balanço energético utilizado para obter o valor das PNTs em sistemas de distribuição compreende três principais etapas: cálculo da PG, estimativa da PT e cálculo da PNT. A Figura 1 apresenta o fluxograma geral do método.

Figura 1 - Balanço de Energia



Fonte: Autora.

- Etapa 1: Cálculo da perda global

O método do balanço de energia é comumente utilizado em sistemas de distribuição quando se deseja determinar as perdas elétricas, pois permite contabilizar o montante de energia perdida (perda global) em todo o processo de distribuição a partir das medições na saída dos alimentadores e do histórico de consumo nas UCs.

A PG do sistema é calculada a partir da diferença entre a energia despachada/fornecida pelo alimentador e a energia faturada nas UCs, conforme mostra a Equação 1.

$$\Delta W_G = \Delta W_{SE} - \Delta W_F \quad (1)$$

Onde ΔW_G é a perda global do sistema (kWh), ΔW_{SE} é a energia fornecida pela subestação (kWh) e ΔW_F é a energia faturada nas unidades consumidoras (kWh).

- Etapa 2: Estimativa da perda técnica

As PTs do sistema de distribuição podem ser estimadas a partir do fluxo de potência da rede. Assim, é possível obter um valor resultante de PT referente às perdas de todos os equipamentos como transformadores, geradores, cargas, cabos etc. O cálculo manual do fluxo de potência é limitado a pequenas redes devido à facilidade de se cometerem erros e ao tempo necessário para realizá-lo. Sendo assim, existem *softwares* disponíveis capazes de calcular o fluxo de potência, como o ANAREDE, DIgSILENT Power Factory, OpenDSS, entre outros.

Nesse trabalho, as PTs das redes de distribuição foram calculadas a partir do *software* OpenDSS, que é muito utilizado pelas concessionárias de energia para cálculo de fluxo de potência. Para obter resultados coerentes de PTs, é necessário que as informações da rede elétrica sejam consistentes.

- Etapa 3: Cálculo da perda não técnica

Uma vez obtido o valor de PG do sistema e o valor estimado de PT, é possível determinar o valor das PNTs, através da Equação 2.

$$W_{PNT} = W_G - W_T \quad (2)$$

Onde W_{PNT} é a perda não técnica (kWh), W_G é a perda global (kWh) e W_T é a perda técnica (kWh).

3.2 DADOS UTILIZADOS

Conforme mencionado anteriormente, os dados necessários para o desenvolvimento desse estudo foram disponibilizados pelo Grupo CPFL Energia. Nas seções subsequentes, serão exibidos esses dados e, em cada seção, será descrito como a metodologia foi aplicada e o tratamento de dados realizado. Vale ressaltar que o algoritmo desenvolvido na linguagem de programação Python realiza o mesmo passo a passo descrito a seguir.

3.2.1 Dados de Faturamento

Os dados de faturamento, que se dividem em grupos A (média tensão) e B (baixa tensão), são fornecidos em formato .csv e consistem em quatro colunas relevantes para o estudo: código da instalação (código da unidade consumidora), data de referência da leitura, número de dias de consumo e quantidade de energia faturada.

O objetivo desta seção é determinar a quantidade de energia faturada por cada alimentador, que é calculada pela soma da energia faturada na média e na baixa tensão, conforme expresso na Equação 3.

$$W_F = W_{F,MT} + W_{F,BT} \quad (3)$$

W_F é a energia total faturada pelo alimentador, $W_{F,MT}$ é a energia faturada pelo alimentador na média tensão e $W_{F,BT}$ é a energia faturada pelo alimentador na baixa tensão.

Para relacionar o consumo das UCs com os seus alimentadores, o Grupo CPFL Energia também fornece um arquivo de cadastro, através do qual é possível determinar a qual alimentador tal unidade consumidora pertence.

3.2.1.1 Tratamento de dados

Devido a diferença entre a data de referência e a data de consumo dos dados de faturamento, foi necessário realizar uma adequação no calendário. A data de referência da leitura indica o dia em que a leitura foi realizada na unidade consumidora, ou seja, em campo.

Já a data do consumo refere-se ao período em que a energia foi utilizada pelo consumidor. Como resultado, nesta metodologia, a data de referência foi alterada para um mês anterior à data de referência original, para se referir à data de consumo.

Além disso, com a análise dos dados recebidos, foram identificados alguns pontos importantes. Para os dados de faturamento, foi observado o número de dias da medição diferente do número de dias do mês de consumo. Então, fez-se necessária a realização da compensação desses dados com base nos dias do calendário, utilizando a Equação 4, onde W_{Fc} é a energia faturada compensada, W_F é a energia faturada no período de consumo, num_dias é o número de dias de consumo e $\Delta dias$ é a diferença de dias entre o número de dias de consumo e o número de dias do mês.

$$W_{Fc} = W_F \pm \left(\Delta dias * \frac{W_F}{num_dias} \right) \quad (4)$$

Por exemplo, se o intervalo de consumo em janeiro compreende 28 dias e o mês de janeiro contém 31 dias, então, é feito o cálculo para adequar o consumo para 31 dias. Considerando um consumo de 220 kWh nos 28 dias, a compensação do consumo para os 31 dias resulta em:

$$W_{Fc} = 220 \text{ kWh} + 3 \text{ dias} * \left(\frac{220 \frac{\text{kWh}}{\text{dias}}}{28 \text{ dias}} \right) = 243,57 \text{ kWh}$$

Se os dias de consumo forem maiores que os dias do calendário, é necessário realizar a adequação, diminuindo a média diária de consumo tantas vezes quanto for a diferença entre os dias. Quando os dias de consumo forem iguais aos do calendário ou estiverem zerados, não é realizada a compensação.

As Tabelas 1 e 2 exemplificam o processo de compensação dos dados de faturamento do Grupo A e B, respectivamente, para o Alimentador A durante o mês de janeiro de 2020. É importante destacar que os dados apresentados nas tabelas possuem como data de referência fevereiro de 2020, ou seja, a compensação é feita para o mês anterior, janeiro, com 31 dias.

Tabela 1 - Compensação de dados de consumo - Grupo A

GRUPO A			
DATA_REFERENCIA	NUM_DIAS	CONSUMO (kWh)	CONSUMO_COMPENSADO (kWh)
202002	0	49172,00	49172,00
202002	0	82366,00	82366,00
202002	0	3731,00	3731,00
202002	0	28112,00	28112,00
202002	0	41010,00	41010,00
202002	0	13666,00	13666,00
202002	0	42378,00	42378,00
202002	0	5151,00	5151,00
202002	0	2930,00	2930,00
202002	0	211475,00	211475,00
202002	0	245846,00	245846,00
202002	0	41890,00	41890,00
202002	0	58795,00	58795,00
202002	0	68188,00	68188,00
202002	0	38156,00	38156,00
202002	0	117975,00	117975,00
202002	0	21731,00	21731,00
202002	0	24580,00	24580,00
202002	0	1324,00	1324,00
202002	0	11333,00	11333,00
202002	0	54951,00	54951,00
202002	0	17521,00	17521,00
202002	0	5441,00	5441,00
202002	0	2840,00	2840,00
202002	0	38719,00	38719,00
202002	0	14095,00	14095,00
202002	0	3769,00	3769,00
202002	0	4258,00	4258,00
202002	0	10379,00	10379,00
202002	0	1326,00	1326,00
CONSUMO TOTAL (kWh)			1263108,00

Fonte: Autora.

Tabela 2 - Compensação de dados de consumo - Grupo B

GRUPO B			
DATA_REFERENCIA	NUM_DIAS	CONSUMO (kWh)	CONSUMO_COMPENSADO (kWh)
202002	28	502	555,79
202002	28	333	368,68
202002	28	65	71,96
202002	28	0	0,00
202002	28	1160	1284,29
202002	28	867	959,89
202002	0	0	0,00
202002	28	1113	1232,25
202002	28	471	521,46
202002	0	0	0,00
202002	0	0	0,00
202002	28	961	1063,96
202002	28	1465	1621,96
202002	28	4422	4895,79
202002	28	833	922,25
202002	0	0	0,00
202002	28	776	859,14
202002	0	0	0,00
202002	28	1722	1906,50
202002	28	2070,5	2292,34
202002	28	150	166,07
202002	0	0	0,00
202002	28	113	125,11
202002	28	192	212,57
202002	28	1364	1510,14
202002	28	347	384,18
202002	28	326	360,93
202002	28	1502	1662,93
202002	28	57	63,11
202002	28	0	0,00
202002	28	302	334,36
202002	28	170	188,21
202002	28	2	2,21
202002	28	30	33,21
202002	28	24	26,57
202002	28	6049	6697,11
202002	28	128	141,71
202002	28	44	48,71
202002	28	465	514,82
202002	28	174	192,64
202002	28	62	68,64
202002	0	0	0,00
202002	28	19	21,04
202002	28	505	559,11
202002	28	237	262,39
202002	28	277	306,68
202002	28	228	252,43
202002	28	78	86,36
202002	28	152	168,29
202002	0	0	0,00
202002	28	82	90,79
202002	28	89	98,54

GRUPO B			
DATA_REFERENCIA	NUM_DIAS	CONSUMO (kWh)	CONSUMO_COMPENSADO (kWh)
202002	28	792	876,86
202002	28	118	130,64
202002	28	11	12,18
202002	0	0	0,00
202002	28	0	0,00
202002	28	331	366,46
202002	28	183	202,61
202002	28	572	633,29
202002	0	0	0,00
202002	28	26	28,79
202002	28	194	214,79
202002	28	5016	5553,43
202002	28	70	77,50
202002	0	0	0,00
202002	0	0	0,00
202002	0	0	0,00
202002	28	6724	7444,43
202002	28	1558	1724,93
202002	28	136	150,57
202002	28	4537,675	5023,85
202002	0	0	0,00
202002	28	2501	2768,96
202002	0	0	0,00
CONSUMO TOTAL (kWh)			58344,41

Fonte: Autora.

De acordo com as Tabelas 1 e 2, pode-se notar que não houve compensação dos dados de faturamento para o Grupo A, já que o número de dias aparece como zero. Por fim, depois de realizar o tratamento dos dados de faturamento, a Tabela 3 apresenta o consumo total do Alimentador A no mês de janeiro de 2020, que foi calculado utilizando a Equação 3.

Tabela 3 - Consumo total 2020/01

ALIMENTADOR A	
GRUPO	CONSUMO (kWh)
Grupo A (MT)	1263108,00
Grupo B (BT)	58344,41
Consumo Total	1321452,41

Fonte: Autora.

3.2.2 Medições SCADA

Os dados das medições na saída dos alimentadores são obtidos através do SCADA e estão contidos em uma única planilha em formato .csv. O registro do fornecimento de energia é feito a cada 15 minutos, durante as 24 horas do dia, resultando em um arquivo com 96 medições, ou seja, 96 pontos. Além disso, o arquivo conta com outras informações relevantes para o estudo como código do alimentador, ano, mês e dia.

Esta seção tem como objetivo determinar a energia total fornecida por cada alimentador mensalmente. Como mencionado anteriormente, os dados de fornecimento consistem em 4 pontos por hora, portanto, a Equação 5 foi utilizada para determinar a energia diária fornecida em kWh.

$$W_{SE, dia}(\text{kWh}) = \frac{\sum_{p=1}^{96} P(\text{kW})_p}{4} \quad (5)$$

Onde $W_{SE, dia}$ é a energia fornecida diária, P é a potência ativa fornecida e p é o ponto de medição.

Para calcular a energia total mensal fornecida pelo alimentador, foi realizado o somatório da energia fornecida diariamente, conforme mostrado na Equação 6.

$$W_{SE, mês}(\text{kWh}) = \sum_{d=1}^{Nd} W_{SE, dia}(\text{kWh})_d \quad (6)$$

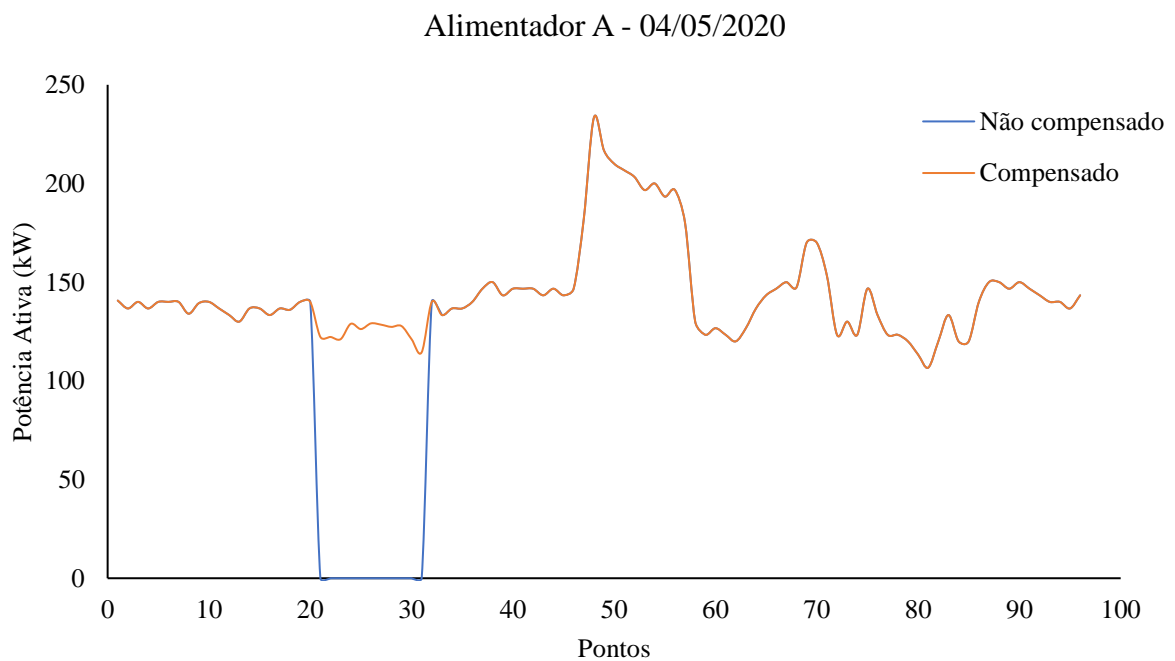
$W_{SE, mês}$ é a energia fornecida pela subestação de energia no mês, $W_{SE, dia}$ é a energia fornecida pela subestação de energia no dia, d é o dia do mês e Nd é o número de dias do calendário.

3.2.2.1 Tratamento de dados

Os dados de fornecimento de energia do SCADA apresentam falhas na medição, ou seja, contém lacunas, o que significa que alguns pontos têm valor zero. Essa falha afeta a estimativa das PGs, resultando em um aumento no erro do cálculo.

Sendo assim, buscando uma forma de compensar os dados em cada ponto zerado, foi utilizado como valor de energia fornecida neste ponto a média aritmética dos demais pontos, considerando o mesmo alimentador, ano, mês e dia da semana. Para exemplificar esse cenário, a Figura 2 apresenta o antes e o depois da compensação dos dados de energia fornecida no dia 04 de maio de 2020 para o Alimentador A.

Figura 2 - Curva diária de potência ativa para o alimentador A em 04 de maio de 2020, antes e após a compensação de medições zeradas



No cenário apresentado, pode-se observar uma lacuna de 10 pontos, preenchida pelo ajuste realizado através da média aritmética descrita anteriormente. Sendo assim, após a compensação dos dados zerados, é possível aplicar as equações descritas ao longo da seção para obter o resultado de energia fornecida total mensal em cada alimentador. Como exemplo, a Tabela 4 traz o resultado da energia fornecida pelo Alimentador A no mês de janeiro de 2020.

Tabela 4 - Energia fornecida total – 2020/01

ALIMENTADOR A	
Energia Fornecida Total (kWh)	878117,70

Fonte: Autora.

3.2.3 Redes Elétricas

As informações das redes elétricas dos alimentadores utilizados neste trabalho foram extraídas do GIS para o *software* OpenDSS.

Os dados são compostos por dez arquivos em formato .dss, sendo um deles denominado "main" ou arquivo principal, que redireciona os demais. Os outros arquivos contêm o registro de cada componente da rede, como linhas, transformadores, capacitores, entre outros. Essa forma de separar os componentes da rede em diferentes arquivos facilita na organização e no entendimento da rede elétrica no OpenDSS e é muito útil quando há a necessidade de alteração na configuração de um dos componentes, como é o caso deste trabalho.

Para o estudo de PT, foi preciso alterar o arquivo de cargas disponível na rede do alimentador, uma vez que se deseja estimar a PT em diferentes períodos. Isso ocorre porque o fluxo de potência nas redes varia para cada período analisado, com alterações na demanda média de cada unidade consumidora do alimentador.

O arquivo de cargas contém informações como o nome da carga, a barra de conexão, número de fases, fator de potência, tensão, potência, entre outros, assim como o registro do transformador ao qual está conectado. Sendo assim, para modificar apenas a informação desejada, neste caso, a demanda média da carga calculada pela Equação 7, arquivos em formato .txt foram criados para cada mês utilizando a função "edit" do OpenDSS, que tem como objetivo editar apenas o componente solicitado. Esses arquivos foram redirecionados a partir do arquivo principal do OpenDSS.

$$D = \frac{W_F \text{ (kWh)}}{Nd * 24h} \quad (7)$$

D é a demanda média da unidade consumidora (em kW), W_F é a energia faturada por unidade consumidora (kWh) conforme cálculo apresentado na seção 3.2.1 e Nd é o número de dias do calendário.

Para estimar as PTs nos alimentadores utilizados neste trabalho, foi adotado o modo de solução diário para o fluxo de potência. Nesse modo, o *software* retorna como solução final a perda diária no alimentador (kWh) que é extraído pelo *energymeter* do OpenDSS, sendo necessário realizar a multiplicação deste valor pelo número de dias do calendário para obter o valor de PT mensal, conforme Equação 8.

$$W_{T_{mensal}} = W_{T_{diária}} * Nd \quad (8)$$

$W_{T_{mensal}}$ é a perda técnica mensal do alimentador, $W_{T_{diária}}$ é a perda técnica diária do alimentador e Nd é o número de dias do calendário.

3.3 IMPLEMENTAÇÃO DA METODOLOGIA

Conforme mencionado nas seções anteriores, um algoritmo em Python foi desenvolvido para automatizar o processo da metodologia de balanço energético apresentada nesse trabalho, pois o estudo realizado manualmente torna-se exaustivo e capaz de propagar erros ao longo do processo, considerando a quantidade de dados utilizados. Esse processo inclui a leitura de arquivos de entrada (arquivos em formato .csv e .dss), tratamento de dados conforme descrito anteriormente para os dados de faturamento e SCADA, operações matemáticas necessárias para a obtenção dos dados de saída de interesse utilizando as equações apresentadas ao longo do trabalho e geração dos dados de saída, que é o objetivo da metodologia.

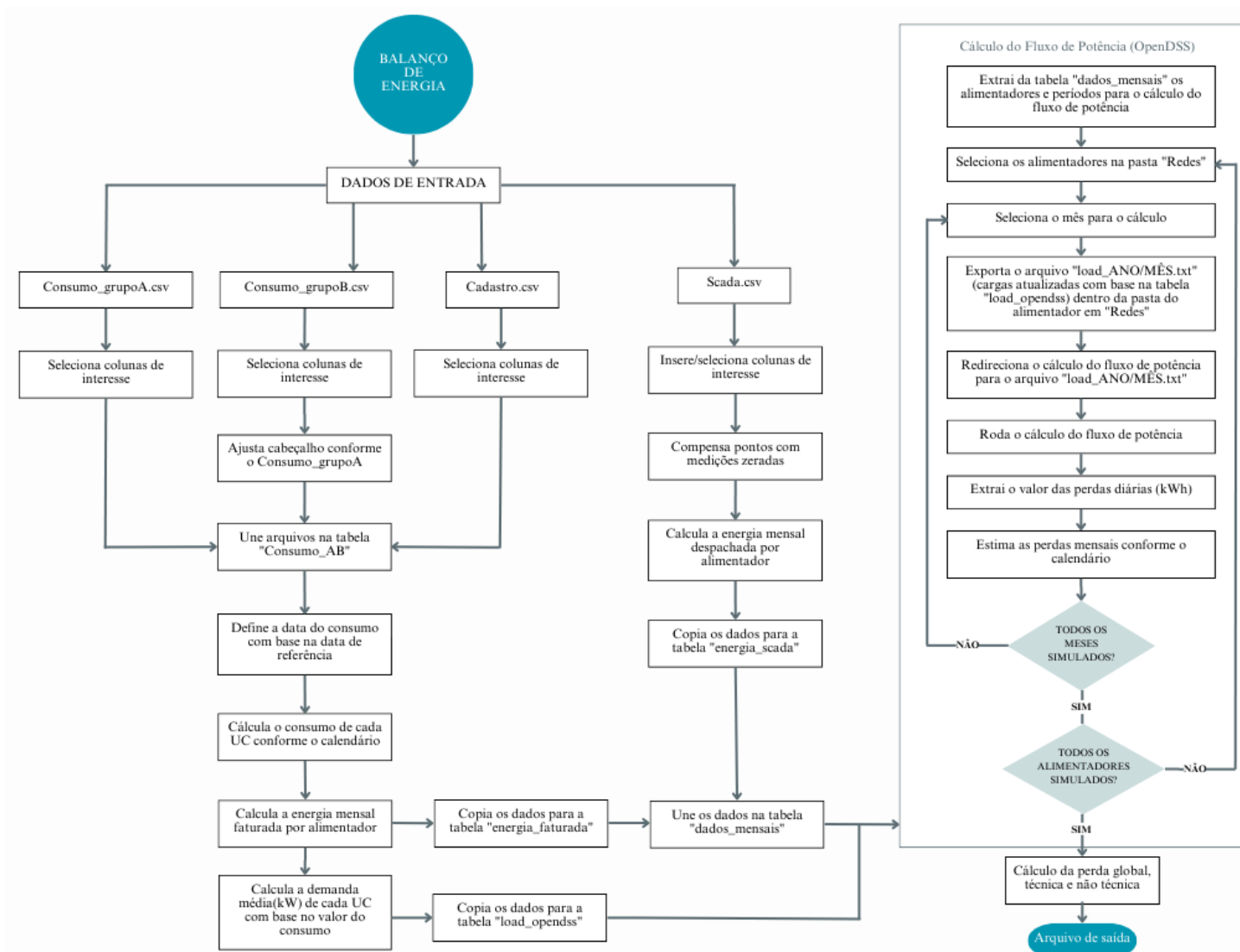
Desta forma, o algoritmo divide-se em etapas, como descritas resumidamente a seguir, que são realizadas para cada alimentador analisado:

- 1) Leitura dos dados de energia fornecida (medições SCADA), tratamento dos dados (lacunas com medições zeradas) e cálculo da energia fornecida total mensal;
- 2) Leitura dos dados comerciais (Grupos A e B), tratamento dos dados (compensação pela data de leitura), cálculo da energia consumida total mensal e cálculo de demanda por unidade consumidora (para alterar os arquivos de carga no OpenDSS);
- 3) Leitura das redes dos alimentadores cadastrados no OpenDSS, atualização de cargas para o período de cálculo desejado e simulações do fluxo de potência (cálculo das PTs);
- 4) Cálculo das PGs mensais;
- 5) Cálculo das PNT mensais;
- 6) Geração do arquivo com os dados de saída.

Para tornar possível a automatização da metodologia no Python, foi necessário a utilização de alguns pacotes, como Numpy e Pandas para a manipulação dos dados em planilhas/tabelas. Além disso, foi utilizado a biblioteca OpenDSSDirect que possibilitou a interface entre Python e OpenDSS para a realização do fluxo de potência nas redes de distribuição obtendo os dados de PTs. A biblioteca OpenDSSDirect não requer que o usuário instale o *software* OpenDSS, o que tornou possível implementar o código na plataforma do Grupo CPFL Energia.

A Figura 3 apresenta um fluxograma mais detalhado do algoritmo desenvolvido no Python. O código desenvolvido na linguagem de programação Python é apresentado no Apêndice A.

Figura 3 - Fluxograma do algoritmo desenvolvido no Python.



Fonte: Autora.

3.3.1 OpenDSS

Para estimar as PTs neste trabalho, foi necessário utilizar um *software* capaz de realizar o cálculo do fluxo de potência das redes de distribuição. Para isso, foi utilizado o OpenDSS a partir da linguagem de programação Python.

A interface entre o OpenDSS e o Python foi possível através da utilização da biblioteca `OpenDSSDirect.py` do Python. O `OpenDSSDirect.py` é uma biblioteca desenvolvida para possibilitar a implementação de uma interface direta do OpenDSS no Python, sem a necessidade de ter o *software* instalado (`OpenDSSDirect.py`).

O OpenDSS é uma ferramenta computacional utilizada para simulação de sistemas elétricos de distribuição de utilidades elétricas (DUGAN; MONTENEGRO, 2020). Esse *software*, é amplamente utilizado em estudos, análise e planejamento de redes com inserção de cargas, geração distribuída fotovoltaica, geradores, baterias e veículos elétricos na rede de distribuição. Para isso, é necessário possuir informações completas e consistentes da rede de distribuição a ser analisada.

O OpenDSS possui vários modos de solução para o fluxo de potência, como modo de solução instantânea, diário, anual, dinâmico, Monte Carlo, entre outros. Quando é usado o modo de solução no qual a carga varia no tempo, é possível determinar o intervalo desejado, podendo ser 1 segundo, 24 horas, 1 mês, e assim por diante (DUGAN; MONTENEGRO, 2020). Para a aplicação da metodologia do balanço energético, as simulações do fluxo de potência nas redes de distribuição foram realizadas no modo diário, com intervalo de 5 em 5 minutos, totalizando 288 pontos no dia (cada hora com 12 pontos).

Quando a simulação do fluxo de potência é finalizada, o usuário pode extrair os resultados de interesse, como as perdas, tensões, potências, entre outros, em cada componente da rede e para o sistema total. Além disso, é possível extrair informações da rede, como número de barras e número de nós (DUGAN; MONTENEGRO, 2020).

Existem diversas formas de cadastrar as cargas no OpenDSS, podendo ser potência ativa (P) constante e potência reativa (Q) constante, modelo de carga com impedância constante (Z), modelo de carga exponencial, modelo de carga zip etc. As cargas não precisam necessariamente ser cadastradas com “*loadshapes*”, ou seja, não precisam de multiplicadores de perfil de carga, pois todos os modelos levam a carga a um perfil de impedância constante com intuito de garantir a convergência mesmo quando a tensão está muito baixa (DUGAN; MONTENEGRO, 2020).

Nesse trabalho, o perfil de carga utilizado foi o modelo zip que descreve a potência ativa e reativa como modelo de carga de potência, corrente e impedância constante, definidos por

coeficientes. As cargas cadastradas no OpenDSS são alteradas de acordo com o período desejado para a solução do fluxo de potência. Assim, buscando alterar apenas a demanda da carga (kW), utilizou-se a função “edit” do OpenDSS, inalterando as demais configurações.

3.3.2 Python

O Python é uma linguagem de programação de alto nível, orientada a objetos e com uma ampla biblioteca de pacotes padrão disponíveis para o usuário. A linguagem foi criada por Guido Van Rossum no ano de 1991 e possui sintaxe simples e é de fácil aprendizado, além de ser totalmente gratuita.

Neste trabalho, a linguagem de programação Python foi escolhida para automatizar o estudo do balanço de energia devido à facilidade de programação para desenvolver rápidas aplicações, análise de dados e rotinas em geral. Além disso, o Python possui a biblioteca OpenDSSDirect.py que possibilita a utilização da interface entre Python e OpenDSS, como já mencionado anteriormente. No estudo realizado, foi necessário utilizar pacotes adicionais como o pandas.py que torna a análise de dados em tabelas muito mais simples.

4 RESULTADOS E DISCUSSÕES

Nesta seção, serão apresentados os resultados do balanço energético para dois estudos de caso realizados como aplicação prática da metodologia. Para isso, como mencionado anteriormente, foram utilizados dados reais de fornecimento, faturamento e de redes elétricas, disponibilizados pelo Grupo CPFL Energia. A Tabela 5 apresenta a relação de número de barras e UCs ativas por alimentador.

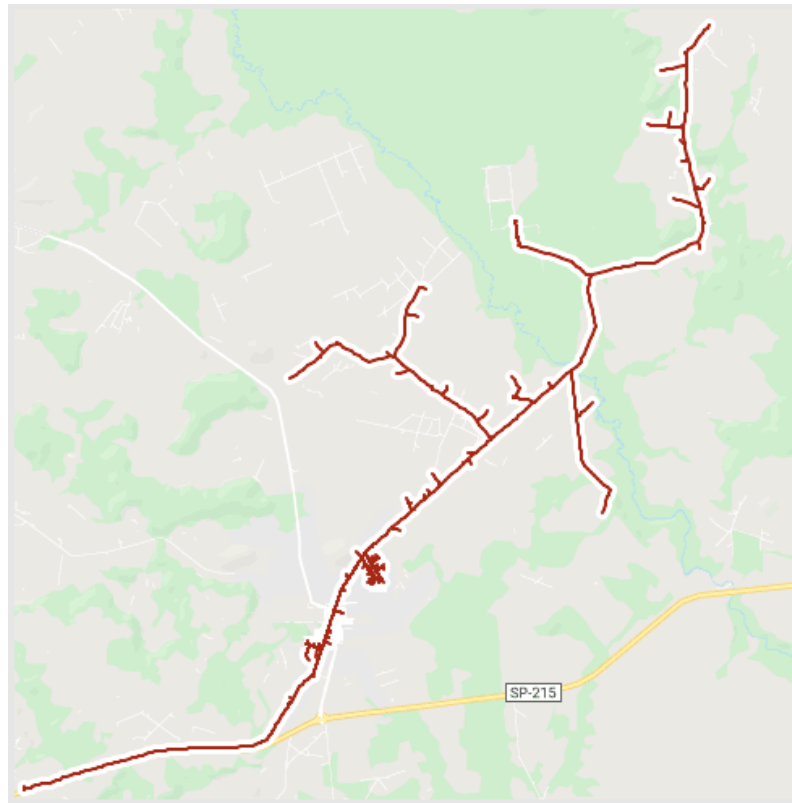
Tabela 5 - Dados dos alimentadores utilizados

Alimentador	Número de Barras	Consumidores Ativos
Alimentador 1	2838	1408
Alimentador 2	5811	2033

Fonte: Autora.

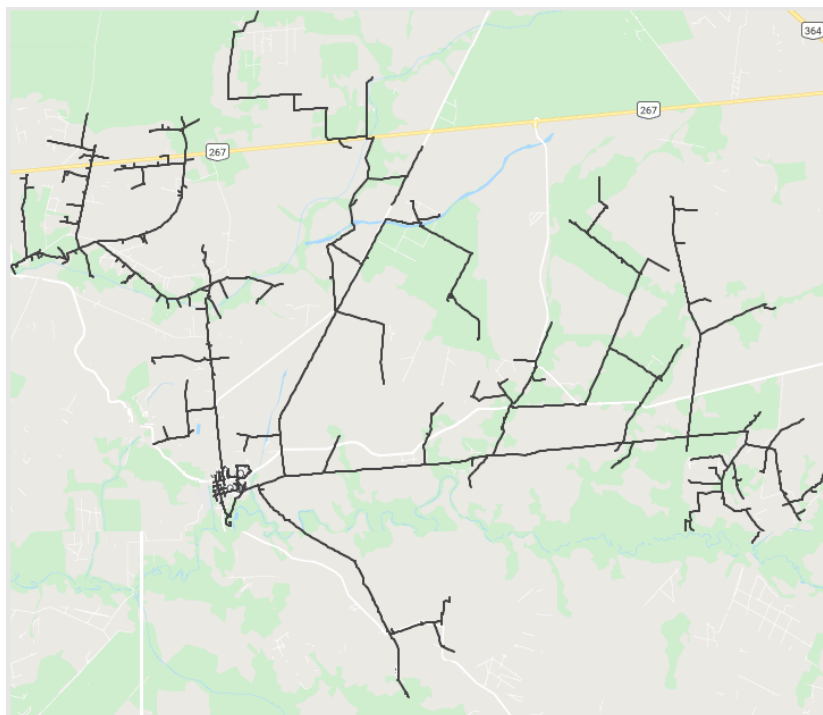
Nas Figuras 4 e 5, é possível observar os diagramas com a topologia das redes elétricas de cada alimentador. É importante ressaltar que ambos os alimentadores utilizados possuem UCs de média e de baixa tensão.

Figura 4 - Topologia da rede elétrica do Alimentador 1 de distribuição



Fonte: KMZ do Google.

Figura 5 - Topologia da rede elétrica do Alimentador 2 de distribuição



Fonte: KMZ do Google.

4.1 RESULTADOS DO CÁLCULO DE PERDA GLOBAL

Os resultados das PGs para ambos os estudos de caso, bem como a energia fornecida e a energia consumida, estão apresentados na Tabela 6. Os valores de perdas são expressos em MWh e em percentual (%). No primeiro caso, o alimentador registra uma PG de energia de 22,52%, o que indica uma área potencial de PNT, que pode estar relacionada à presença de consumidores ilegais conectados ao sistema, roubo ou furto de energia. No segundo caso, o alimentador apresenta um valor de PG consideravelmente menor, com 9,94%, indicando poucas chances de haver PNT por fraudes na rede.

Tabela 6 - Resultado da Perda Global para os Alimentadores 1 e 2

		Energia Fornecida	Energia Consumida	Perda Global
Alimentador 1	MWh	816,79	632,81	183,98
	%	100,00	77,48	22,52
Alimentador 2	MWh	1167,39	1051,29	116,09
	%	100,00	90,06	9,94

Fonte: Autora.

4.2 RESULTADO DA ESTIMATIVA DAS PERDAS TÉCNICAS.

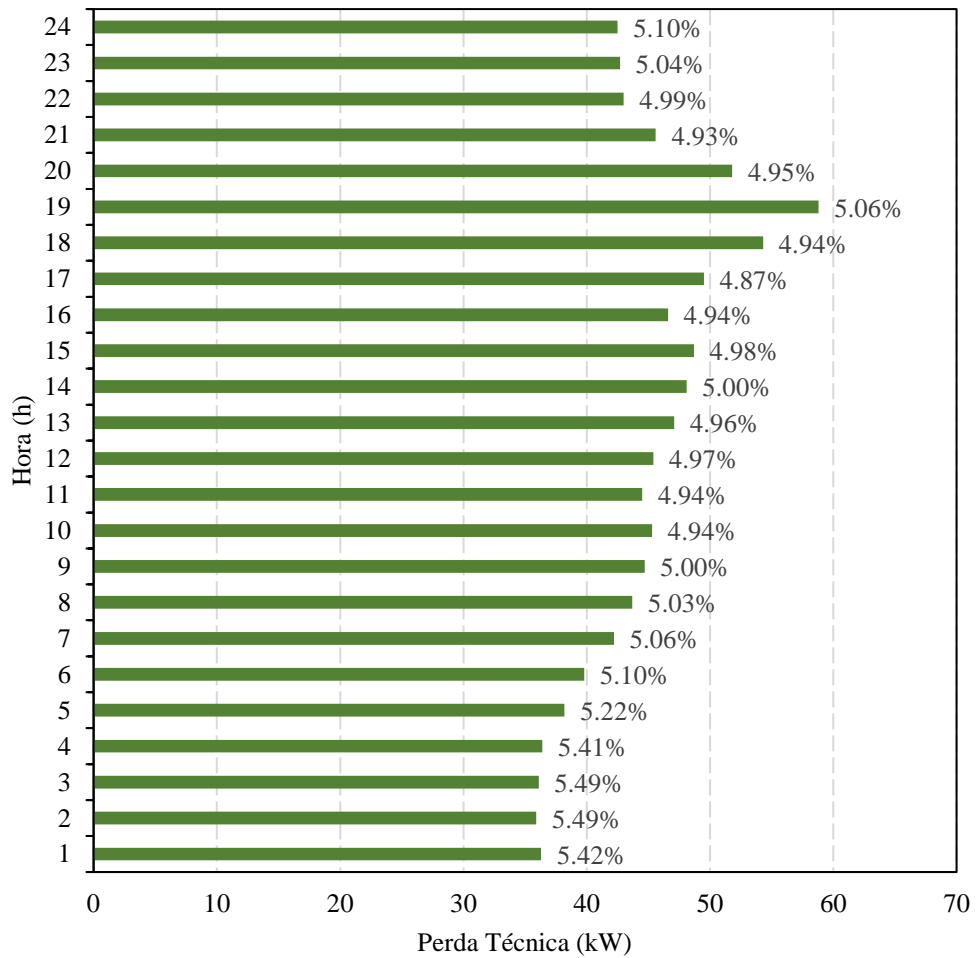
As Figuras 6 a 9, apresentam a estimativa das PTs diárias e da potência ativa fornecida pelo alimentador da subestação para os casos 1 e 2, respectivamente. A simulação foi realizada para cada hora do dia, utilizando o OpenDSS.

Nas Figuras 6 e 7, referentes ao Alimentador 1, é perceptível pelo perfil de PT que a carga predominante é de natureza comercial e residencial. Isso se deve à maior demanda de potência ativa fornecida pela subestação durante o período das 7h às 21h (indicando a característica comercial) e ao aumento na demanda das 18h às 21h (característica residencial). Já no Alimentador 2, nas Figuras 8 e 9, o perfil da curva de potência ativa fornecida pela subestação indica que a carga predominante nesse alimentador é de natureza comercial, com maior demanda das 7h às 21h.

Adicionalmente, as Figuras 6 e 8 exibem a porcentagem de perda em relação à potência fornecida pela subestação para cada hora dia, mantendo-se em torno de 5% de perda para ambos

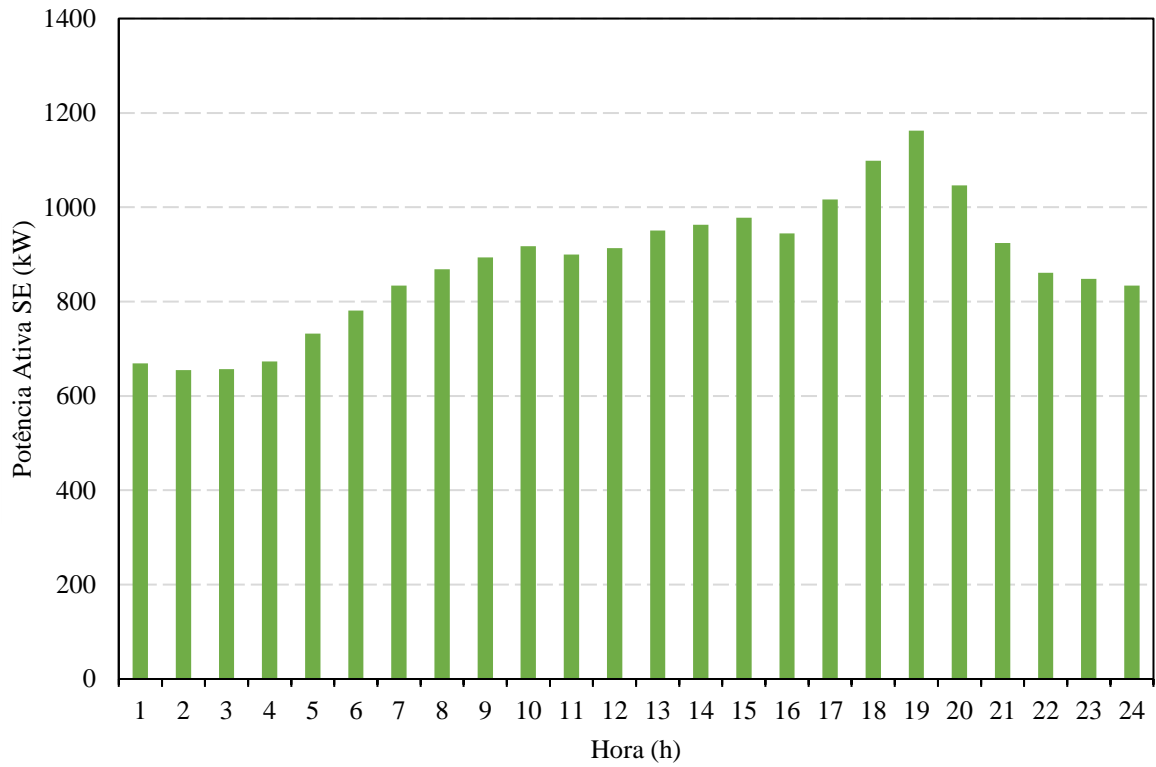
os casos. O resultado das PTs mensais para cada alimentador é apresentado na Tabela 7, resultando em 4,05% de PT para o Alimentador 1 e 4,46% de PT para o Alimentador 2.

Figura 6 - Resultado de Perda Técnica no período de 24h para o Alimentador 1



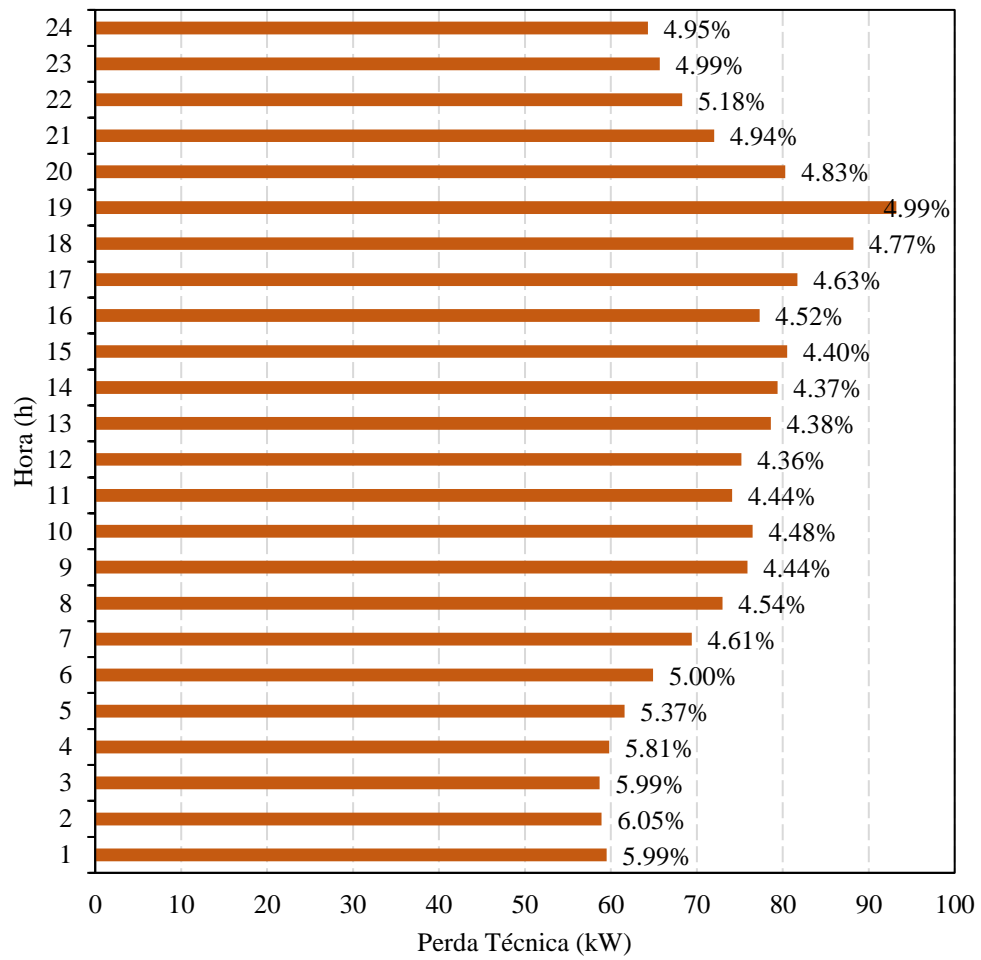
Fonte: Autora.

Figura 7 - Potência Ativa Fornecida pelo Alimentador 1 na SE no período de 24h



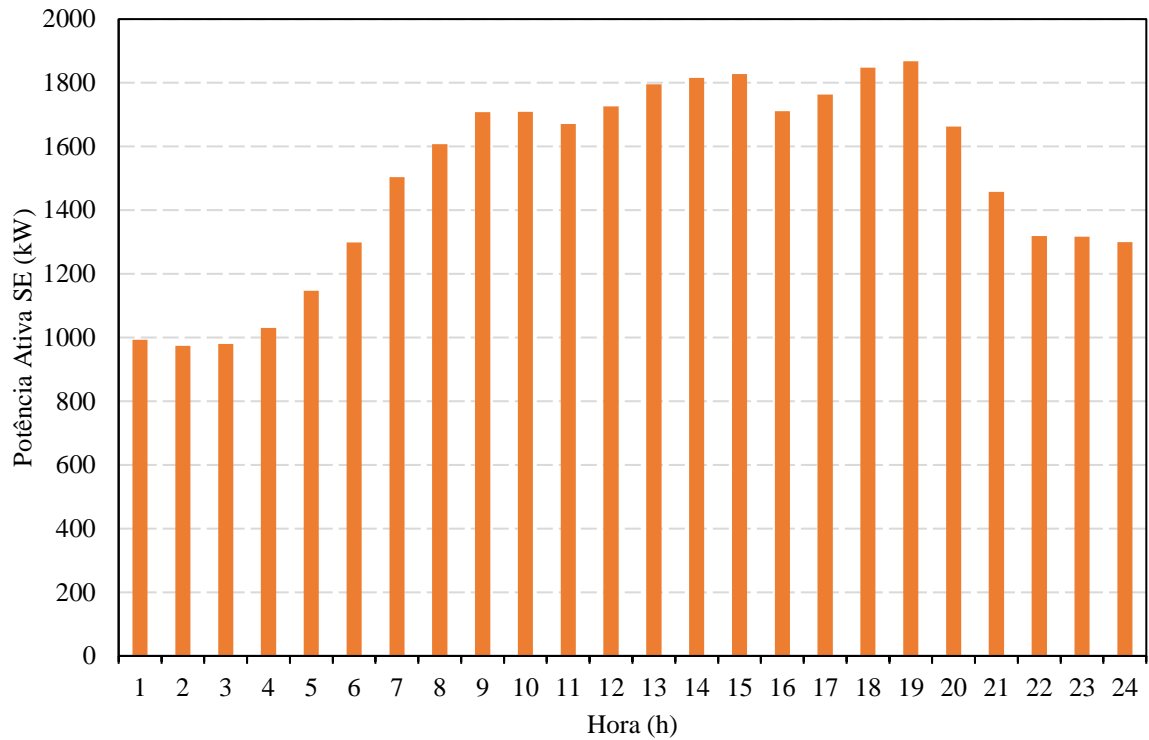
Fonte: Autora.

Figura 8 - Resultado de Perda Técnica no período de 24h para o Alimentador 2



Fonte: Autora.

Figura 9 - Potência Ativa Fornecida pelo Alimentador 2 na SE no período de 24h



Fonte: Autora.

Tabela 7 - Resultado de Perda Técnica mensal para os Alimentadores 1 e 2

	Perda Técnica Mensal (MWh)	%
Alimentador 1	33,05	4,05
Alimentador 2	52,02	4,46

Fonte: Autora.

4.3 RESULTADO DAS PERDAS NÃO TÉCNICAS

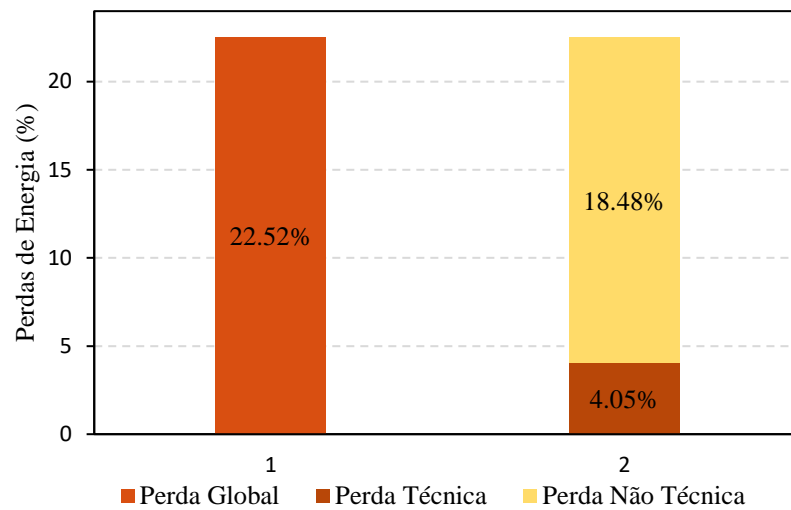
Por fim, os resultados da metodologia do balanço energético apresentam os valores das PNTs nos sistemas de distribuição analisados, conforme mostrado na Tabela 8. Além disso, nas Figuras 10 e 11, é possível observar as porcentagens de PGs, PTs e PNT em relação à energia fornecida, para ambos os estudos.

Tabela 8 - Resultado de Perda Não Técnica para os Alimentadores 1 e 2

	Perda Global	Perda Técnica	Perda Não Técnica
Alimentador 1 (MWh)	183,98	33,05	150,92
Alimentador 2 (MWh)	116,09	52,02	64,07

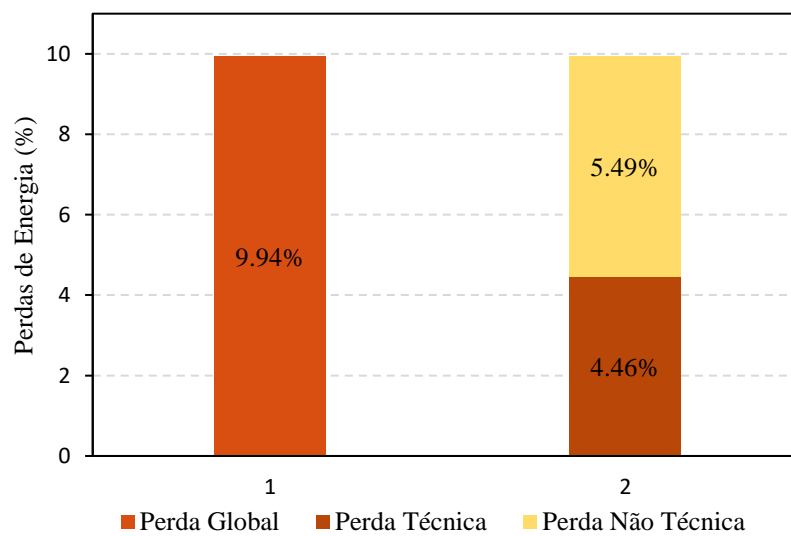
Fonte: Autora.

Figura 10 - Percentual de PG, PT e PNT da energia fornecida pelo Alimentador 1 de distribuição



Fonte: Autora.

Figura 11 - Percentual de PG, PT e PNT da energia fornecida pelo Alimentador 2 de distribuição



Fonte: Autora.

Com base nos resultados apresentados nas Figuras 10 e 11, é possível observar dois cenários diferentes para os estudos de caso realizados. Inicialmente, de acordo com a Figura 10, o primeiro estudo de caso revela que 22,52% da energia fornecida pelo Alimentador 1 corresponde a PGs. Dentro desse valor, 18,48% são atribuídos a PNT no sistema, um percentual relativamente alto, o que indica a possibilidade de haver consumidores ilegais conectados à rede e/ou ocorrência de furtos de energia por parte dos próprios clientes. Por outro lado, no segundo estudo de caso, a Figura 11 mostra que a PG para o Alimentador 2 foi de 9,94%, sendo que 5,49% desse valor refere-se às PNT, indicando poucas chances de haver fraudes nesse alimentador devido ao baixo percentual de perda. É importante considerar que esse percentual pode estar relacionado a erros intrínsecos dos medidores, erros na leitura e faturamento da energia, entre outros fatores.

Além disso, é relevante destacar que as PNTs não estão diretamente ligadas ao tamanho da rede elétrica. Diversos fatores podem influenciar a porcentagem de PNTs em alimentadores, como a região e o tipo de consumidores conectados, tais como residenciais, comerciais, industriais e rurais. Por outro lado, as PTs podem ter relação com o tamanho da rede, uma vez que quanto maior a extensão da rede e o número de componentes, maior é a probabilidade de ocorrerem perdas técnicas ao longo do sistema, como pode ser observado na Tabela 7, onde o Alimentador 2 apresentou maior valor de PT em comparação ao Alimentador 1.

Os resultados das PNTs para os alimentadores analisados por meio dessa metodologia servem como indicadores para determinar a região e os alimentadores onde as PNTs ocorrem, mas não são resultados definitivos. É importante ressaltar que mesmo alimentadores com baixo percentual de perda podem ter a presença de consumidores ilegais conectados, porém em menor escala.

5 CONCLUSÃO

O trabalho realizado envolveu o desenvolvimento de um algoritmo em Python para realizar o balanço de energia em sistemas de distribuição. Esse algoritmo utiliza dados reais de faturamento e fornecimento de energia, além de informações das redes elétricas. Por meio do OpenDSS, o algoritmo calcula o fluxo de potência nas redes elétricas e determina o valor das PGs no sistema analisado durante o período desejado. Os resultados obtidos incluem as PGs, PTs e, principalmente, as PNTs nos sistemas de distribuição.

Adicionalmente, neste trabalho, foram realizadas aplicações práticas da metodologia por meio de dois estudos de caso, nos quais foi possível observar dois tipos de cenários. Em um deles, foi identificado um alto percentual de PNTs, indicando a provável presença de UCs irregulares. No outro cenário, foi constatado um baixo percentual de PNT, indicando uma baixa probabilidade de as UCs estarem em situação irregular.

Os resultados obtidos neste trabalho foram satisfatórios, uma vez que o algoritmo desenvolvido em Python é capaz de realizar o processo de balanço de energia com análise e tratamento de dados, incluindo os cálculos necessários conforme o método estabelecido. Essa abordagem resultou em uma ferramenta simples e eficaz para a detecção de PNT.

Além disso, essa metodologia faz parte de um projeto maior que tem como objetivo determinar as PNTs e identificar onde elas ocorrem por meio da utilização de inteligência artificial (IA). Os dados resultantes dessa metodologia são utilizados para detectar possíveis regiões e alimentadores com PNT, sendo empregados como indicadores e informações de entrada para dar suporte à IA.

A utilização da biblioteca OpenDSSDirect do Python permitiu a implementação do algoritmo desenvolvido na plataforma do Grupo CPFL Energia. Essa implementação será utilizada pela concessionária, juntamente com a IA, para detectar PNTs nos sistemas de distribuição da empresa. Isso representa um avanço significativo na capacidade de análise e monitoramento do sistema elétrico, contribuindo para a eficiência na redução de perdas de energia não técnicas.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGENCIA NACIONAL DE ENERGIA ELÉTRICA (ANEEL). SIGA - Sistema de Informações de Geração da ANEEL. ANEEL. Disponível em: <<https://app.powerbi.com/view?r=eyJrIjoibjI4OGYyYjQtYWM2ZC00YjllLWJlYmEtYzdkNTQ1MTc1NjM2IiwidCI6IjQwZDZmOWI4LWVjYTctNDZhMi05MmQ0LWVhNGU5YzAxNzBlMSIsImMiOjR9>>. Acesso em: 10 maio 2023.
- AGÜERO, J. R. Improving the efficiency of power distribution system through technical and nontechnical losses reduction. **IEEE PES Transmission and Distribution Conference and Exposition**, Orlando, FL, USA, May 2012. 8.
- BERNARDON, D. P. et al. Perdas Técnicas e Comerciais de Energia Elétrica em Sistemas de Distribuição. **VII Conferência Brasileira de Qualidade de Energia Elétrica**, Santa Maria, 2007. 1-6.
- BIAZZI, R. R. et al. Technical and Nontechnical Energy Loss Estimation Including Volt/Var. **Journal of Control, Automation and Electrical Systems**, 2021. 1-13.
- BIAZZI, R. R. et al. TECHNICAL AND NONTECHNICAL LOSS ESTIMATION USING REAL-TIME ACCESS TO DISTRIBUTION SYSTEM. **CIREN 2021 - The 26th International Conference and Exhibition on Electricity Distribution**, 2021. 5.
- BRETAS, A. S. et al. Distribution networks nontechnical power loss estimation: A hybrid datadriven physics model-based framework. **Electric Power Systems Research**, September 2020. 10.
- DAVIDSON, I. E. Evaluation and effective management of nontechnical losses in power networks. **Transactions of the South African Institute of Electrical Engineers**, September 2003. 39-42.
- DUGAN, R. C.; MONTENEGRO, D. **Reference Guide: The Open Distribution System**. Electric Power Research Institute, Inc: OpenDSS. [S.l.], p. 1-242. 2020.
- GARCIA, T. R. et al. Evaluation of Energy Losses Using Smart Meter Data. **2021 Workshop on Communication Networks and Power Systems (WCNPS)**, Brasilia, Brazil, 2021. 1-6.
- KASSEM, H. E.; BADR, M.; AHMED, S. A. Reduction of energy losses in electrical distribution systems. **24th International Conference on Electricity Distribution (CIREN)**, Stockholm, 2013. 1-4.
- LEITE, J. B.; MANTOVANI, J. R. S. Detecting and locating non-technical losses in modern distribution networks. **IEEE Transactions on Smart Grid**, March 2018. 1023-1032.
- MADRIGAL, M.; RICO, J. J.; UZCATEGUI, L. Estimation of Non-Technical Energy Losses in Electrical Distribution Systems. **IEEE Latin America Transactions**, 2017. 6.
- OPENDSSDIRECT.PY. **Python Package Index (PyPI)**. Disponível em: <<https://pypi.org/project/OpenDSSDirect.py/>>. Acesso em: 22 Maio 2023.
- ROSSINI, A. et al. Contribution to distribution systems technical and nontechnical losses estimations using WLG state estimator. **IEEE Power & Energy Society General Meeting**, Chicago, IL, USA, July 2017. 1-5.

SILVA, A. F. D. METODOLOGIA PARA ANÁLISE DO BALANÇO ENERGÉTICO USANDO ESTIMADOR DE ESTADOS E MEDIÇÕES EM TEMPO REAL.

Universidade Federal de Santa Maria - UFSM. Santa Maria, p. 1-99. 2018.

SILVEIRA, M. F. D. et al. **Sistemas Elétricos de Potência**. 1. ed. [S.l.]: Sagah Educação S.A., 2022. Disponível em:

<<https://integrada.minhabiblioteca.com.br/reader/books/9786556900872/pageid/19>>. Acesso em: 10 maio 2023.

TREVIZAN, R. D.; BRETAS, A. S.; ROSSINI, A. Distribution Test System for Nontechnical Loss Detection. **2018 North American Power Symposium (NAPS)**, Fargo, ND, USA, 2018. 1-6.

APÊNDICE A

```

import opendssdirect as dss
import pandas as pd
import numpy as np
import datetime
from datetime import timedelta
from calendar import import monthrange
from datetime import date
import os

pd.set_option('display.max_row', 5000)
pd.set_option('display.max_columns', 5000)
pd.set_option('display.width', 5000)

class DadosScada:
    def __init__(self, var_scada):
        self.input_scada = var_scada

    def leitura_scada_modelo_1(self):

        df_scada = pd.read_csv(f'{self.input_scada}', low_memory=False,
sep=';', encoding="ISO-8859-1")
        df_scada.drop_duplicates(subset=['ALIMENTADOR', 'ANO', 'MES',
'DIA'], keep='first', inplace=True)

        df_scada.rename(columns={'ALIMENTADOR': 'COD_ALIMENTADOR'},
inplace=True)
        df_scada['ANO'] = df_scada['ANO'].astype(int)
        df_scada['MES'] = df_scada['MES'].astype(int)
        df_scada['DIA'] = df_scada['DIA'].astype(int)
        df_scada['DIAS_CALENDARIO'] = df_scada.apply(lambda row:
monthrange(int(row['ANO']), int(row['MES']))[1],
axis=1)

        for ponto in range(1, 97):
            try:
                df_scada[f'PONTO_{ponto}'] =
df_scada[f'PONTO_{ponto}'].astype(float)
            except ValueError:
                df_scada[f'PONTO_{ponto}'] =
df_scada[f'PONTO_{ponto}'].apply(
lambda row: float(str(row).replace(',','.')))

        return df_scada

    def clc_energia_scada(self):

        df_scada = self.input_scada.copy()

        df_grouped = df_scada.groupby(['COD_ALIMENTADOR', 'ANO', 'MES',
'DIAS_CALENDARIO'])
        df_grouped = df_grouped['PONTO_1'].agg([np.size]).reset_index()
        df_missing_data = df_grouped.loc[df_grouped['DIAS_CALENDARIO'] !=
df_grouped['size']].reset_index()
        df_size = len(df_missing_data.index)
        if df_size > 0:
            for index in range(0, df_size):
                line = df_missing_data.loc[[index], ['COD_ALIMENTADOR',

```

```

'ANO', 'MES', 'DIAS_CALENDARIO']]
    alimentador = list(line['COD_ALIMENTADOR'])[0]
    ano = int(line['ANO'])
    mes = int(line['MES'])
    dias_calendario_int = int(line['DIAS_CALENDARIO'])

    df_filtred = df_scada.loc[(df_scada['COD_ALIMENTADOR'] ==
alimentador) &
                                (df_scada['ANO'] == ano) &
(df_scada['MES'] == mes)]

    dias_medidos = set(df_filtred['DIA'])
    dias_calendario = {x for x in range(1, dias_calendario_int
+ 1)}

    dias_faltantes = dias_calendario.difference(dias_medidos)
    df_dias_faltantes = pd.DataFrame({'COD_ALIMENTADOR':
alimentador, 'ANO': ano, 'MES': mes,
                                'DIA': [x for x in
dias_faltantes]})
    df_scada = pd.merge(df_dias_faltantes, df_scada,
how='outer')

    for ponto in range(1, 97):
        df_scada[f'PONTO_{ponto}'].fillna(0, inplace=True)

    df_scada['DIA_SEMANA'] = df_scada.apply(
        lambda row:
datetime.datetime.weekday(pd.to_datetime(str(row['ANO']) + '-' +
str(row['MES']) + '-' +
str(row['DIA']))), axis=1)
    df_scada['DIAS_CALENDARIO'] = df_scada.apply(lambda row:
monthrange(int(row['ANO']), int(row['MES'])))[1],
axis=1)
    df_scada['total_pontos'] = df_scada.apply(lambda row:
row['DIAS_CALENDARIO'] * 96, axis=1)

    df = pd.melt(df_scada.reset_index(),
                id_vars=['COD_ALIMENTADOR', 'ANO', 'MES',
'DIA_SEMANA', 'DIAS_CALENDARIO', 'total_pontos'],
                var_name='PONTO',
                value_vars=[f'PONTO_{x}' for x in range(1, 97)],
value_name='kW')

    df_scada_original = df.groupby(['COD_ALIMENTADOR', 'ANO', 'MES',
'DIAS_CALENDARIO', 'total_pontos'])
    df_scada_original = df_scada_original['kW'].agg([np.count_nonzero,
np.sum]).reset_index()
    df_scada_original[f'pontos_scada(i)'] =
df_scada_original.apply(lambda row: round(row['count_nonzero'], 2),
axis=1)
    df_scada_original[f'ENERGIA_SCADA(i)'] =
df_scada_original.apply(lambda row: round(row['sum'] / 4, 2), axis=1)
    df_scada_original.drop(columns=['count_nonzero', 'sum'],
inplace=True)

    df_grouped_01 = df.groupby(['COD_ALIMENTADOR', 'ANO', 'MES',
'DIA_SEMANA', 'PONTO'])
    df_grouped_01 = df_grouped_01['kW'].agg([np.size, np.count_nonzero,

```

```

np.sum]).reset_index()
    df_grouped_01['media_01'] = df_grouped_01.apply(lambda row: 0 if
row['count_nonzero'] == 0 else round(row['sum'] / row['count_nonzero'], 2),
axis=1)
    df_grouped_01['add'] = df_grouped_01.apply(lambda row: row['size']
- row['count_nonzero'], axis=1)
    df_grouped_01['sum(kw)'] = df_grouped_01.apply(lambda row:
row['sum'] + row['media_01'] * row['add'], axis=1)
    df_grouped_01['ptos_com_medicao'] = df_grouped_01.apply(lambda row:
row['size'] if row['media_01'] > 0 else row['count_nonzero'], axis=1)

    df_scada_final = df_grouped_01.groupby(['COD_ALIMENTADOR', 'ANO',
'MES']).agg({'sum(kw)': [np.sum], 'ptos_com_medicao':
[np.sum]}).reset_index()
    df_scada_final.columns = [''.join(col) for col in
df_scada_final.columns]
    df_scada_final.rename(columns={'sum(kw) sum': 'ENERGIA_SCADA(KWH)',
'ptos_com_medicaosum': 'pontos_scada(f)'}, inplace=True)
    df_scada_final['ENERGIA_SCADA(KWH)'] = df_scada_final.apply(lambda
row: round(row['ENERGIA_SCADA(KWH)'] / 4, 2), axis=1)

    df_scada_out = pd.merge(df_scada_original, df_scada_final,
how='outer', on=['COD_ALIMENTADOR', 'ANO', 'MES'])
    df_scada_out = df_scada_out[['COD_ALIMENTADOR', 'ANO', 'MES',
'DIAS_CALENDARIO',
                                'total_pontos', 'pontos_scada(i)',
'pontos_scada(f)',
                                'ENERGIA_SCADA(i)',
'ENERGIA_SCADA(KWH)']]
    df_scada_out.reset_index()

    periodo = df_scada_out[['COD_ALIMENTADOR', 'ANO', 'MES',
'DIAS_CALENDARIO']]
    return df_scada_out, periodo

class DadosComerciais:
    def __init__(self, var_comercial):
        self.input_comercial = var_comercial

    def leitura_regiao(self):
        df_regiao = pd.read_csv(f'{self.input_comercial}', sep=';',
encoding="ISO-8859-1", low_memory=False)
        df_regiao.rename(columns={'UC': 'COD_INSTALACAO'}, inplace=True)

        return df_regiao

    def leitura_cadastro(self):
        df_cadastro = pd.read_csv(f'{self.input_comercial}', sep=';',
encoding="ISO-8859-1", low_memory=False)
        df_cadastro = df_cadastro.drop_duplicates()
        df_cadastro = df_cadastro[['COD_SISTEMA_SETORIAL',
'DESC_SISTEMA_SETORIAL', 'COD_INSTALACAO', 'COD_ALIMENTADOR',
                                'DESC_EMPRESA']]

        return df_cadastro

    def leitura_grupoA(self):
        df_grupo_a = pd.read_csv(f'{self.input_comercial}', sep=';',
encoding="ISO-8859-1")

```



```

df_grupo_a.fillna(0, inplace=True)
df_grupo_a.drop_duplicates(subset=['COD_INSTALACAO',
'DT_REFERENCIA'], keep='first', inplace=True)

df_grupo_a['NUM_DIAS'] = df_grupo_a['NUM_DIAS'].astype(int)
try:
    df_grupo_a['CONSUMO'] = df_grupo_a['CONSUMO'].astype(float)
except ValueError:
    df_grupo_a['CONSUMO'] = df_grupo_a['CONSUMO'].apply(lambda row:
float(str(row).replace(',','.')))
return df_grupo_a

def leitura_grupoB(self):

    df_grupo_b = pd.read_csv(f'{self.input_comercial}',
low_memory=False, sep=';', encoding="ISO-8859-1")
    df_grupo_b.fillna(0, inplace=True)
    df_grupo_b = df_grupo_b[['COD_INSTALACAO', 'DT_REFERENCIA',
'NUM_DIAS', 'QTD_ENERGIA_FATURADA']]
    df_grupo_b.drop_duplicates(subset=['COD_INSTALACAO',
'DT_REFERENCIA'], keep='first', inplace=True)
    df_grupo_b.rename(columns={'QTD_ENERGIA_FATURADA': 'CONSUMO'},
inplace=True)

    df_grupo_b['NUM_DIAS'] = df_grupo_b['NUM_DIAS'].astype(int)
    try:
        df_grupo_b['CONSUMO'] = df_grupo_b['CONSUMO'].astype(float)
    except ValueError:
        df_grupo_b['CONSUMO'] = df_grupo_b['CONSUMO'].apply(lambda row:
float(str(row).replace(',','.')))

    return df_grupo_b

def clc_energia_consumo(self, df_perodo_scada):

    df_consumo = self.input_comercial.copy()

    df_consumo['ANO'] = df_consumo['DT_REFERENCIA'].apply(lambda row:
int(str(row)[0:4]))
    df_consumo['MES'] = df_consumo['DT_REFERENCIA'].apply(lambda row:
int(str(row)[4:6]))
    df_consumo['DIA'] = [1 for x in range(0, len(df_consumo))]
    df_consumo['DATA_CONSUMO'] = (df_consumo[['ANO', 'MES',
'DIA']]).apply(lambda row: date(row['ANO'], row['MES'],
row['DIA']).replace(day=1) - timedelta(days=1), axis=1)
    df_consumo['ANO'] = df_consumo['DATA_CONSUMO'].apply(lambda row:
int(row.year))
    df_consumo['MES'] = df_consumo['DATA_CONSUMO'].apply(lambda row:
int(row.month))

    df_consumo = pd.merge(df_perodo_scada, df_consumo, how='inner',
on=['COD_ALIMENTADOR', 'ANO', 'MES'])

    df_consumo['consumo_est'] = (df_consumo[['CONSUMO', 'NUM_DIAS',
'DIAS_CALENDARIO']]).apply(lambda row: round(row['CONSUMO'], 2) if
row['NUM_DIAS'] == 0 else round(row['CONSUMO'] + (row['DIAS_CALENDARIO'] -
row['NUM_DIAS']) * (row['CONSUMO'] / row['NUM_DIAS']), 2), axis=1)
    df_consumo['kW'] = (df_consumo[['consumo_est', 'NUM_DIAS',
'DIAS_CALENDARIO']]).apply(lambda row: round(row['consumo_est'] / (24 *
row['DIAS_CALENDARIO']), 4), axis=1)

```

```

        df_irr = df_consumo.loc[df_consumo['COD_SISTEMA_SETORIAL'] == 414]
        df_consumo_irr = df_irr.groupby(['COD_ALIMENTADOR', 'DESC_EMPRESA',
'ANO', 'MES']).agg({'CONSUMO': [np.sum], 'consumo_est': [np.sum],
'COD_INSTALACAO': [np.size]}).reset_index()
        df_consumo_irr.columns = [''.join(col) for col in
df_consumo_irr.columns]
        df_consumo_irr.rename(columns={'CONSUMOsum':
'ENERGIA_FATURADA_IRRIG(i)',
'consumo_estsum':
'ENERGIA_FATURADA_IRRIGANTES(KWH)',
'COD_INSTALACAOsize':
'NUM_IRRIGANTES'}, inplace=True)

        df_consumofat_out = df_consumo.groupby(['COD_ALIMENTADOR',
'DESC_EMPRESA', 'ANO', 'MES']).agg({'CONSUMO': [np.sum],
'consumo_est': [np.sum],
'COD_INSTALACAO': [np.size]}).reset_index()
        df_consumofat_out.columns = [''.join(col) for col in
df_consumofat_out.columns]
        df_consumofat_out.rename(columns={'CONSUMOsum':
'ENERGIA_FATURADA(i)',
'consumo_estsum':
'ENERGIA_FATURADA(KWH)',
'COD_INSTALACAOsize':
'NUM_CLIENTES'},
inplace=True)

        df_consumofat_out = pd.merge(df_consumofat_out, df_consumo_irr,
how='left')

        df_load_opendss = df_consumo[['COD_INSTALACAO', 'COD_ALIMENTADOR',
'DT_REFERENCIA', 'ANO', 'MES',
'CONSUMO', 'NUM_DIAS', 'consumo_est',
'kW']]

        return df_consumofat_out, df_load_opendss

class DSS:

    def __init__(self, end_dss):
        self.end_dss = end_dss

    if dss.Basic.Start(0) is False:
        print("Problema em iniciar o OpenDSS")

    def compila_DSS(self):

        dss.run_command("compile " + self.end_dss)

    def redirect_load(self):
        dss.run_command(f"Redirect loads")

    def redirect_newload(self, dta):
        dss.run_command(f"Redirect loads_{dta}.txt ")

```

```

def solve_snap(self):
    dss.run_command(f"Set Mode=snapshot")
    dss.run_command("Solve")

def solve_mod0(self):
    dss.run_command(f"Set Mode=daily time=(0,3300) number=288
stepsize=5m")
    dss.run_command("Set ControlMode=Time")
    dss.run_command("Solve")

def get_all_load_names(self):
    return dss.Loads.AllNames()

def kw_load(self, loadnames):
    load_kw_list = []
    for load in loadnames:
        dss.Loads.Name(load)
        load_kw_list.append(dss.Loads.kW())
    return load_kw_list

def perdas(self):
    return round(dss.Circuit.Losses()[0] / 1000, 4)

def get_meters(self):
    energia_sub = (dss.Meters.Totals()[0])
    energia_carga = (dss.Meters.Totals()[4])
    energia_pt = (dss.Meters.Totals()[12])
    return energia_sub, energia_carga, energia_pt

def perda_tecnica(df_mensal, df_demanda_ucs, df_redes):
    df_demanda_ucs['periodo'] = (df_demanda_ucs[['ANO',
'MES']]).apply(lambda row: str(row['ANO']) + '_' + str(row['MES']), axis=1)
    df_demanda_ucs = df_demanda_ucs[['COD_INSTALACAO', 'COD_ALIMENTADOR',
'ANO', 'MES', 'kW', 'periodo']]

    al_list = []
    periodo_list = []
    pt_list = []
    ef_list = []
    ec_list = []

    if len(df_redes) > 0:

        df_solve =
df_redes.groupby(['COD_ALIMENTADOR']).agg(list).reset_index()
        dic_tosove = {list(df_solve['COD_ALIMENTADOR'])[index]:
list(df_solve['periodo'])[index]
                    for index in range(0, len(df_solve.index))}
        df_demanda_ucs = pd.merge(df_redes, df_demanda_ucs, how='left',
on=['COD_ALIMENTADOR', 'periodo'])
        df_demanda_ucs['COD_INSTALACAO'] =
df_demanda_ucs['COD_INSTALACAO'].astype(str)
        cwd_original = os.getcwd()
        for al, periodo in dic_tosove.items():
            print(f'\n\nAlimentador: {al}')
            objeto = DSS(fr"redes/{al}/Master.dss")
            objeto.compila_DSS()
            objeto.solve_snap()
            loadfullnames_list = objeto.get_all_load_names()

```

```

cwd_redes = os.getcwd()
if len(loadfullnames_list) == 0:
    os.chdir(cwd_original)
    continue
print(f'    Período de simulação:')
for dta in periodo:
    print(f'        {dta.split("_")[0]}/{dta.split("_")[1]}')

    df_filtred =
df_demanda_ucs.loc[(df_demanda_ucs['COD_ALIMENTADOR'] == al) &
                    (df_demanda_ucs['periodo']
== dta)].reset_index()
    df_filtred = df_filtred[['COD_INSTALACAO', 'kW']]
    uc_comercial = set(df_filtred['COD_INSTALACAO'])

    txt_df = pd.DataFrame()
    txt_df['COD_INSTALACAO'] = loadfullnames_list
    txt_df = pd.merge(txt_df, df_filtred, how='left')
    txt_df.fillna(0, inplace=True)

    with open(fr"loads_{dta}.txt", "w") as arquivo:
        for index in range(0, len(txt_df.index)):
            arquivo.write(f"Edit
Load.{list(txt_df['COD_INSTALACAO'])[index]} "
                        f"kW={list(txt_df['kW'])[index]} "
                        f"model=8 ZIPV=(0.50, 0.00, 0.50,
1.00, 0.00, 0.00, 0.5)\n")

    cadastrar_uc_list =
list(uc_comercial.difference(set(loadfullnames_list)))
    cadastrar_uc_df = pd.DataFrame()
    cadastrar_uc_df['COD_INSTALACAO'] = cadastrar_uc_list
    if len(cadastrar_uc_list) > 0:
        cadastrar_uc_df = pd.merge(cadastrar_uc_df, df_filtred,
how='left')
        cadastrar_uc_df.to_csv(fr"Newload_{dta}.csv",
index=False, sep=',')
    else:
        cadastrar_uc_df.to_csv(fr"Newload_{dta}.csv",
index=False, sep=',')

    objeto.redirect_newload(dta)
    objeto.solve_mod()
    energia_fornecida, energia_consumida, perda_tenica_kwh =
objeto.get_meters()
    al_list.append(al)
    periodo_list.append(dta)
    ef_list.append(energia_fornecida)
    ec_list.append(energia_consumida)
    pt_list.append(perda_tenica_kwh)

    os.chdir(cwd_redes)
    os.chdir(cwd_original)

df_fluxo_potencia = pd.DataFrame()
df_fluxo_potencia['COD_ALIMENTADOR'] = al_list
df_fluxo_potencia['periodo'] = periodo_list
df_fluxo_potencia['EF_ESTIMADA'] = ef_list
df_fluxo_potencia['EC_ESTIMADA'] = ec_list
df_fluxo_potencia['PT(KWH)'] = pt_list

```

```

df_fluxo_potencia = pd.merge(df_mensal, df_fluxo_potencia, how='outer',
on=['COD_ALIMENTADOR', 'periodo'])
df_fluxo_potencia.fillna(0, inplace=True)
df_fluxo_potencia['EF_ESTIMADA'] = df_fluxo_potencia.apply(lambda row:
round(row['EF_ESTIMADA'] * row['DIAS_CALENDARIO'], 2), axis=1)
df_fluxo_potencia['EC_ESTIMADA'] = df_fluxo_potencia.apply(lambda row:
round(row['EC_ESTIMADA'] * row['DIAS_CALENDARIO'], 2), axis=1)
df_fluxo_potencia['PT(KWH)'] = df_fluxo_potencia.apply(lambda row:
round(float(row['PT(KWH)']) * int(row['DIAS_CALENDARIO']), 2), axis=1)

return df_fluxo_potencia

def clc_perda_mensal(df_kwh_scada, df_kwh_faturado, df_fluxo_potencia):
df_mensal = pd.merge(df_kwh_scada, df_kwh_faturado, how='inner')
df_mensal = pd.merge(df_mensal, df_fluxo_potencia, how='left')

df_mensal['PG(KWH)'] = df_mensal.apply(lambda row:
row['ENERGIA_SCADA(KWH)'] - row['ENERGIA_FATURADA(KWH)'], axis=1)
df_mensal['dif_1'] = df_mensal.apply(lambda row: row['total_pontos'] -
row['pontos_scada(i)'], axis=1)
df_mensal['dif_2'] = df_mensal.apply(lambda row: row['total_pontos'] -
row['pontos_scada(f)'], axis=1)
df_mensal['flag1'] = df_mensal.apply(
lambda row: col_flag1(row['total_pontos'], row['pontos_scada(i)'],
row['ENERGIA_SCADA(i)'], row['dif_1'],
row['dif_2']), axis=1)
df_mensal['ENERGIA_ESTIMADA'] = df_mensal.apply(lambda row:
round(row["ENERGIA_SCADA(KWH)"] - row["ENERGIA_SCADA(i)"], 2) if
row['flag1'] == 'scada-compensado*' or row['flag1'] == 'scada-compensado'
else '', axis=1)
df_mensal['PONTOS_ESTIMADOS'] = df_mensal.apply(lambda row:
round(row["pontos_scada(f)"] - row["pontos_scada(i)"], 2) if row['flag1']
== 'scada-compensado*' or row['flag1'] == 'scada-compensado' else '',
axis=1)

df_mensal['flag2'] = df_mensal.apply(lambda row:
col_flag2(row['PG(KWH)'], row['flag1']), axis=1)

df_mensal['PNT(KWH)'] = df_mensal.apply(lambda row: row['PG(KWH)'] -
row['PT(KWH)'] if row['flag2'] == '' else '',
axis=1)
df_mensal['PG(%)'] = df_mensal.apply(lambda row: round(row['PG(KWH)'] /
row['ENERGIA_SCADA(KWH)'] * 100, 2)
if row['flag2'] == '' else '', axis=1)
df_mensal['PT(%)'] = df_mensal.apply(lambda row: round(row['PT(KWH)'] /
row['ENERGIA_SCADA(KWH)'] * 100, 2)
if row['flag2'] == '' else '', axis=1)
df_mensal['PNT(%)'] = df_mensal.apply(lambda row: round(row['PNT(KWH)']
/ row['ENERGIA_SCADA(KWH)'] * 100, 2)
if row['flag2'] == '' else '', axis=1)
df_mensal['INDICE_PNT'] = df_mensal.apply(lambda row:
col_indice_pnt(row['PNT(%)']), axis=1)

df_mensal['DATA'] = df_mensal.apply(lambda row: str(row['MES']) + '/' +
str(row['ANO']), axis=1)

df_mensal = df_mensal[['COD_ALIMENTADOR', 'DESC_EMPRESA', 'DATA',
'ENERGIA_SCADA(KWH)', 'ENERGIA_ESTIMADA',
'PONTOS_ESTIMADOS',
'ENERGIA_FATURADA(KWH)'],

```

```

'ENERGIA_FATURADA_IRRIGANTES(KWH) ',
    'NUM_CLIENTES', 'NUM_IRRIGANTES',
    'PG(KWH) ', 'PT(KWH) ', 'PNT(KWH) ',
    'PG(%) ', 'PT(%) ', 'PNT(%) ', 'INDICE_PNT',
    'flag1', 'flag2']]

df_mensal.rename(columns={'ENERGIA_SCADA(KWH)': 'ENERGIA_MEDIDA(KWH)',
    'NUM_CLIENTES': 'QUANTIDADE_CLIENTES',
    'NUM_IRRIGANTES': 'QUANTIDADE_IRRIGANTES',
    'PG(KWH)': 'PERDA_GLOBAL(KWH)',
    'PT(KWH)': 'PERDA_TECNICA(KWH)',
    'PNT(KWH)': 'PERDA_NAO_TECNICA(KWH)',
    'PG(%)': 'PERDA_GLOBAL(%)',
    'PT(%)': 'PERDA_TECNICA(%)',
    'PNT(%)': 'PERDA_NAO_TECNICA(%)',
    'flag1': 'FLAG1', 'flag2': 'FLAG2'},

inplace=True)
return df_mensal

def col_flag1(total_pontos, pontos_scada_i, energia_scada_i, dif_1, dif_2):
    if energia_scada_i < 0:
        return 'scada-negativo'
    elif dif_2 / total_pontos * 100 >= 5:
        return fr'scada-incompleto-{{pontos_scada_i}}/{{total_pontos}}'
    elif dif_1 > 0:
        if dif_2 == 0:
            return 'scada-compensado'
        elif 0 < dif_2 / total_pontos * 100 <= 5:
            return 'scada-compensado*'
    else:
        return ''

def col_flag2(pg, flag1):
    if pg < 0:
        return 'inconclusivo'
    elif 'incompleto' in flag1 or 'negativo' in flag1:
        return 'inconclusivo'
    else:
        return ''

def col_indice_pnt(pnt):
    if pnt == '':
        return ''
    else:
        if 0 < pnt <= 15.00:
            return 1
        elif 15.00 < pnt < 20.00:
            return 2
        elif 20.00 < pnt < 25.00:
            return 3
        elif 25.00 < pnt < 30.00:
            return 4
        elif 30.00 < pnt:
            return 5

def define_fluxo_potencia(df_mensal):
    pasta = r"redes"

```

```

subpasta_list = []
for diretorio, subpastas, arquivos in os.walk(pasta):
    subpasta_list.extend(subpastas)
df_redes = pd.DataFrame()
df_redes['COD_ALIMENTADOR'] = subpasta_list

df_mensal['periodo'] = df_mensal.apply(lambda row: str(row['ANO']) +
'_' + str(row['MES']), axis=1)

df_redes_out = pd.merge(df_redes, df_mensal, how='inner',
on=['COD_ALIMENTADOR'])
df_redes_out = df_redes_out[['COD_ALIMENTADOR', 'periodo']]

return df_redes_out

print(f'\nCÁLCULO DA ENERGIA FORNECIDA POR ALIMENTADOR')
print('    * leitura das entradas')
file_scada = DadosScada(fr'scada.csv')
file_df = file_scada.leitura_scada_modelo_1()

print('    * ajuste dos dados')
dataframe_scada = DadosScada(file_df)
energia_scada_df, periodo_scada_df = dataframe_scada.clc_energia_scada()

print(f'\nCÁLCULO DA ENERGIA FATURADA')
file_regiao = DadosComerciais("relatorio.csv")
file_cadastro = DadosComerciais(fr"cadastro.csv")
file_consumoA = DadosComerciais(fr"consumo_grupoA.csv")
file_consumoB = DadosComerciais(fr"consumo_grupoB.csv")

print('    * leitura das entradas')
cadastro_df = file_cadastro.leitura_cadastro()
grupoA_df = file_consumoA.leitura_grupoA()
grupoB_df = file_consumoB.leitura_grupoB()

print('    * ajuste dos dados')
consumo_AB_df = pd.merge(grupoA_df, grupoB_df, how='outer')
consumo_AB_df = pd.merge(consumo_AB_df, cadastro_df, how='left')

dataframe_consumo = DadosComerciais(consumo_AB_df)
energia_faturada_df, load_opendss_df =
dataframe_consumo.clc_energia_consumo(periodo_scada_df)

print(f'\nCLC FLUXO DE POTÊNCIA\n')
dados_mensais_df = pd.merge(energia_scada_df, energia_faturada_df,
how='inner', on=['COD_ALIMENTADOR', 'ANO', 'MES'])
redes_opendss_df = define_fluxo_potencia(dados_mensais_df)
fluxo_potencia_df = perda_tecnica(dados_mensais_df, load_opendss_df,
redes_opendss_df)

perda_mensal_df = clc_perda_mensal(energia_scada_df, energia_faturada_df,
fluxo_potencia_df)

perda_mensal_df.to_csv(fr"out/perda_mensal.csv", sep=';', index=False)

```

NUP: 23081.069129/2023-80

Prioridade: Normal

Homologação de ata de defesa de TCC e estágio de graduação
125.322 - Bancas examinadoras de TCC: indicação e atuação

COMPONENTE

Ordem	Descrição	Nome do arquivo
10	Trabalho de conclusão de curso (TCC) (125.32)	TCC II_Eduarda Becker.pdf

Assinaturas

19/06/2023 14:50:54

DIEGO BERLEZI RAMOS (PROFESSOR DO MAGISTÉRIO SUPERIOR)
07.09.02.00.0.0 - CURSO DE ENGENHARIA ELÉTRICA - CEELE



Código Verificador: 2857822

Código CRC: 870109f8

Consulte em: <https://portal.ufsm.br/documentos/publico/autenticacao/assinaturas.html>

