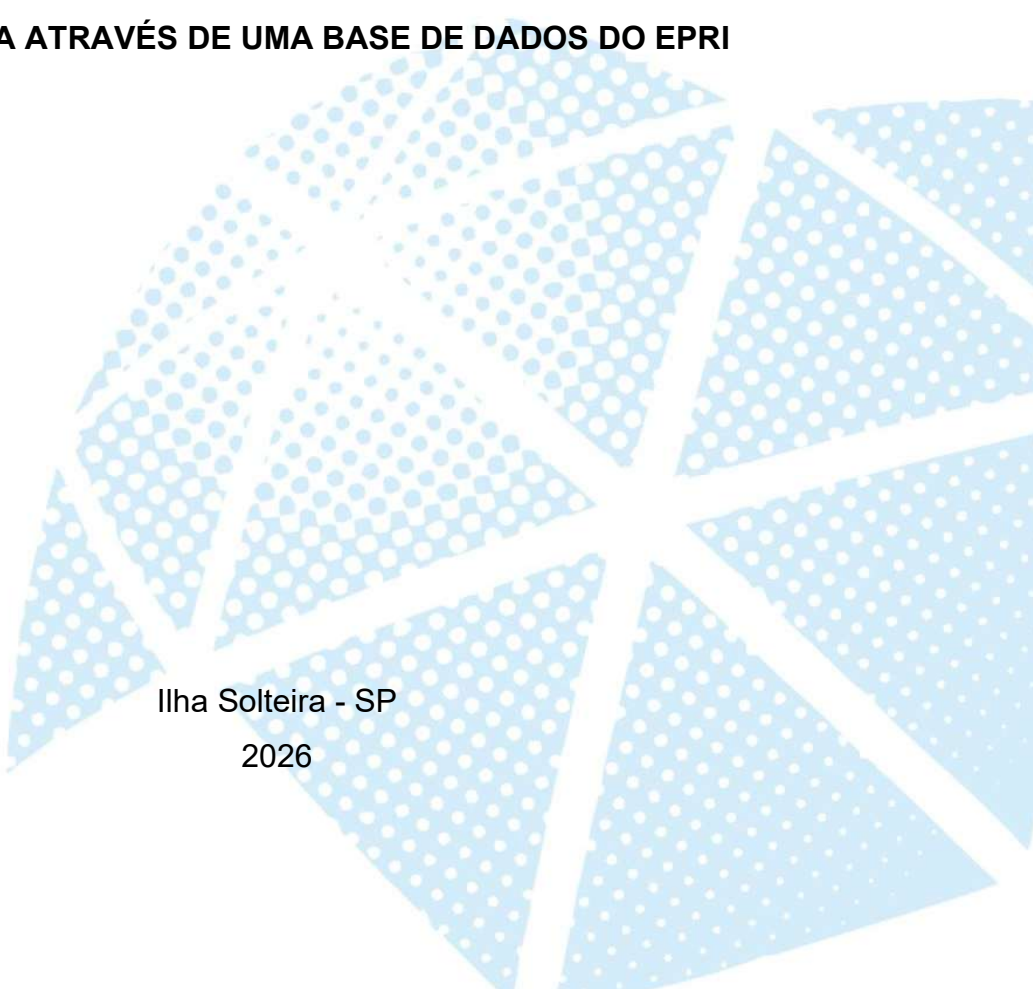


**UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
CAMPUS DE ILHA SOLTEIRA**

MATHEUS DE PAULA FERREIRA

**ANÁLISE DO ALGORITMO DE FLUXO DE POTÊNCIA COM ESTRUTURA PA-
DRONIZADA ATRAVÉS DE UMA BASE DE DADOS DO EPRI**

Ilha Solteira - SP
2026



PROGRAMA DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MATHEUS DE PAULA FERREIRA

ANÁLISE DO ALGORITMO DE FLUXO DE POTÊNCIA COM ESTRUTURA PADRONIZADA ATRAVÉS DE UMA BASE DE DADOS DO EPRI

TCC apresentado(a) à Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, Ilha Solteira, para obtenção do título de Grau acadêmico Bacharel em Engenharia Elétrica.

Área de Concentração: Conforme consta no certificado de aprovação

Orientador(a): Prof. Dr. Jonas Boas Leite

Ilha Solteira - SP

2026

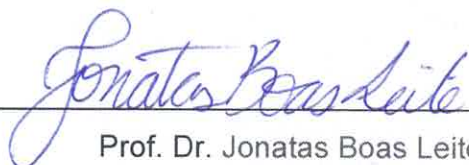
FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

- F383a Ferreira, Matheus de Paula.
Análise do algoritmo de fluxo de potência com estrutura pa-dronizada através de uma base de dados do EPRI / Matheus de Paula Ferreira. -- Ilha Solteira: [s.n.], 2026
65 f. : il.
- Trabalho de conclusão de curso (Graduação em Engenharia Elétrica) - Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, Ilha Solteira, 2026
- Orientador: Jonatas Boas Leite
- Inclui bibliografia
1. Fluxo de potência trifásico. 2. Common information model. 3. CiM/DCIM. 4. Sistemas de distribuição. 5. OpenDSS. 6. Varredura Backward/Forward.

ATA DE DEFESA DE TRABALHO DE GRADUAÇÃO

Aos vinte e seis dias do mês de junho do ano de dois mil e vinte e seis, o discente **Matheus de Paula Ferreira**, matriculado sob o nº182053741, tendo como banca examinadora o seu orientador, o Prof. Dr. Jonatas Boas Leite, o Mestrando Eric Felipe Garcia Souza e o Mestrando Mikael de Freitas Amaral, apresentou o Trabalho de Graduação intitulado "Análise do Algoritmo de Fluxo de Potência com Estrutura Padronizada através de uma Base de Dados do EPRI", obtendo a nota 10,0 (DEZ) e conceito APROVADO.



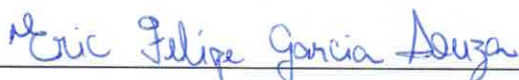
Prof. Dr. Jonatas Boas Leite

- Orientador -



Matheus de Paula Ferreira

- Discente -



Mestrando Eric Felipe Garcia Souza

- Membro da Banca -



Mestrando Mikael de Freitas Amaral

- Membro da Banca -

SUMÁRIO

DEDICATÓRIA	4
AGRADECIMENTOS	5
EPIÍGRAFE	6
RESUMO 7	
ABSTRACT	8
LISTA DE FIGURAS	9
LISTA DE TABELAS	10
LISTA DE QUADROS	11
LISTA DE ABREVIATURAS E SIGLAS	12
LISTA DE SÍMBOLOS	13
1 INTRODUÇÃO	14
1.1 CONTEXTO	14
1.2 MOTIVAÇÃO	15
1.3 OBJETIVOS	16
1.3.1 Objetivo geral.....	16
1.3.2 Objetivos específicos.....	16
1.4 ORGANIZAÇÃO DO TRABALHO	16
2 FLUXO DE POTÊNCIA TRIFÁSICO.....	18
2.1 ALGORITMO DE VARREDURA <i>BACKWARD/FORWARD</i>	18
2.2 REPRESENTAÇÃO COMPUTACIONAL DA REDE	19
2.3 BUSCA EM PROFUNDIDADE RECURSIVA	20
2.4 CRITÉRIO DE CONVERGÊNCIA	22
3 MODELO DE INFORMAÇÃO COMUM (CIM/DCIM)	24
3.1 DEFINIÇÃO E ABRANGÊNCIA.....	24
3.2 PACOTES RELEVANTES PARA O FLUXO DE POTÊNCIA EM DISTRIBUIÇÃO.....	25
3.3 EXTENSÃO DO CIM COM ATRIBUTOS PRIVADOS.....	27
3.4 SERIALIZAÇÃO RDF/XML E A BIBLIOTECA TC57CIM.DLL	28
4 METODOLOGIA: IMPLEMENTAÇÃO EM C#.....	30
4.1 VISÃO GERAL DA ARQUITETURA.....	30
4.2 A BIBLIOTECA TC57CIM.DLL	33
4.3 PARSER CIM E CORREÇÃO DO <i>LOCALE</i>	33

4.4	EXTRAÇÃO DE POTÊNCIA POR BARRA	34
4.5	CÁLCULO DA MATRIZ DE IMPEDÂNCIA 3×3 DAS LINHAS	35
4.6	MODELAGEM DE TRANSFORMADORES COMO RAMO FÍSICO.....	36
4.6.1	Caminho A: TransformerMeshImpedance	37
4.6.2	Caminho B: TransformerTank com ensaios	37
4.7	ADAPTAÇÃO DA DFS PARA INCORPORAR TRANSFORMADORES...	38
4.8	TRADUÇÃO CIM → REDE	40
4.9	RELATÓRIOS EM DISCO.....	41
5	RESULTADOS E COMPARAÇÃO COM OPENDSS	43
5.1	VALIDAÇÃO NO SISTEMA-TESTE DE 135 BARRAS.....	43
5.2	CIRCUITO CKT7 DO EPRI: ANÁLISE DETALHADA.....	44
5.2.1	Características do circuito	44
5.2.2	Procedimento de execução	45
5.2.3	Estatísticas globais	45
5.2.4	Perfil de tensão do primário	47
5.2.5	Distribuição das diferenças percentuais.....	48
5.2.6	Dispersão entre os dois simuladores.....	49
5.2.7	Curva de convergência	50
5.2.8	Discussão dos desvios.....	51
5.2.9	Corrente de ramo.....	53
5.3	CIRCUITOS CKT5 E CKT24: SÍNTESE.....	55
6	CONCLUSÃO	58
	REFERÊNCIAS.....	62

DEDICATÓRIA

Aos meus pais, pela base; ao meu orientador, pela bússola.

AGRADECIMENTOS

À Universidade Estadual Paulista (UNESP), campus de Ilha Solteira, e ao corpo docente do curso de Engenharia Elétrica, pela infraestrutura e excelência na formação acadêmica.

Ao meu orientador, Prof. Dr. Jonas Boas Leite, pela confiança, direcionamento técnico e paciência fundamentais para o desenvolvimento deste trabalho.

Aos pesquisadores do Laboratório de Planejamento de Sistemas de Energia Elétrica (LAPSEE), em especial aos autores dos trabalhos que antecederam e fundamentaram esta pesquisa, cujas contribuições serviram de alicerce para as implementações aqui propostas.

Aos meus pais e familiares, pelo suporte incondicional durante todos os anos de graduação

EPÍGRAFE

"Se você torturar os dados por tempo suficiente, eles confessarão qualquer coisa." — Ronald Coase

RESUMO

A interoperabilidade entre sistemas de informação no setor de distribuição de energia elétrica depende de modelos padronizados como o Common Information Model (CIM), normatizado pela International Electrotechnical Commission. Apresenta-se, neste trabalho, a implementação e a validação experimental de um algoritmo de fluxo de potência trifásico para sistemas de distribuição modelados em CIM/DCIM, com base no perfil CIM 100 do Electric Power Research Institute (EPRI). O algoritmo emprega a varredura *backward/forward* em coordenadas de fase, formalizada por Leite e Mantovani (2015), e estende o motor desenvolvido por Ferreira (2021) e Souza (2023) com quatro contribuições próprias: a derivação direta das grandezas eletricamente úteis a partir do CIM bruto; a modelagem do transformador como ramo físico da rede, com a relação de transformação incorporada à busca em profundidade; a substituição do critério de convergência em potência por um critério em tensão, indispensável em grandezas físicas do Sistema Internacional; e a correção de um erro de localização regional (*locale*) na leitura de atributos numéricos do CIM. A validação foi conduzida sobre os três circuitos de teste do EPRI — ckt5, ckt7 e ckt24 —, com análise estatística detalhada do ckt7 em tensão e em corrente, frente ao simulador OpenDSS. Os resultados indicam boa concordância entre os dois simuladores, com leve tendência de superestimação da tensão pelo programa desenvolvido, atribuída a uma diferença de modelagem no equivalente da subestação: o programa mantém a tensão nominal na barra da fonte, ao passo que o OpenDSS considera a queda sob carga em impedância declarada a montante. A corrente de ramo apresentou concordância semelhante nos três circuitos avaliados, confirmando a validade do algoritmo operando em grandezas físicas do SI. As limitações remanescentes e as direções de pesquisa futura são apresentadas no Capítulo 6.

Palavras-chave: Fluxo de Potência Trifásico. *Common Information Model*. CIM/DCIM. Sistemas de Distribuição. OpenDSS. Varredura *Backward/Forward*.

ABSTRACT

Interoperability between information systems in the electric power distribution sector has come to depend on the adoption of standardized models such as the Common Information Model (CIM), specified by the International Electrotechnical Commission. This work presents the implementation and the experimental validation of a three-phase power flow algorithm for distribution systems modeled in CIM/DCIM, based on the CIM 100 profile adopted by the Electric Power Research Institute (EPRI). The algorithm uses the *backward/forward* sweep method in its three-phase, phase-coordinates formulation by Leite and Mantovani (2015), and extends the engine developed by Ferreira (2021) and Souza (2023) with four original contributions: the direct derivation of the electrically useful quantities from the raw CIM data (the Z_{abc} matrix of the lines, the S vector per bus, and the physical parameters of the transformers); the modelling of the transformer as a physical branch of the network, incorporating the transformation ratio a into the depth-first traversal of the radial tree; the replacement of the convergence criterion $|\Delta S| < \varepsilon$ by $\max|\Delta V| < \text{tol}_V$, which is essential for operation in physical SI units; and the confinement of the `CultureInfo.InvariantCulture` to the scope of the CIM parser, which prevents the misinterpretation of the decimal separator under a Brazilian Portuguese *locale*. The validation was carried out on the three EPRI test circuits (ckt5, ckt7 and ckt24), with detailed statistical analysis of circuit ckt7, in both voltage and current, against the OpenDSS simulator output. On the 12.47 kV primary of ckt7, the computed voltage shows a median difference of +0.93% relative to OpenDSS, with all readings within the range of +0.30% to +2.86%; the program stays slightly above the simulator because it holds the substation equivalent at the nominal voltage (1.0 pu), whereas the OpenDSS `vsource`, declared at 1.05 pu behind an impedance, has its terminal lowered to about 0.969 pu under the nominal load. The branch current magnitude agrees with a median of $\approx 3.8\%$ on the primary and $\approx 5.4\%$ over the set of lines, within the same range observed for ckt5 (3.8%) and ckt24 (5.0%). The results confirm the validity of the sweep algorithm operating in physical SI units. The remaining limitations, among them the modelling of the `EnergySource` as a Thévenin branch and the representation of voltage-regulation controls, and the future research directions are presented in Chapter 6.

Keywords: Three-Phase Power Flow. Common Information Model. CIM/DCIM. Distribution Systems. OpenDSS. *Backward/Forward Sweep*.

LISTA DE FIGURAS

Figura	Título
Figura 2.1	Hierarquia de classes do modelo elétrico
Figura 3.1	Hierarquia simplificada das classes CIM consumidas pelo <i>parser</i>
Figura 3.2	Extensão do CIM com atributos privados de resultado
Figura 5.1	Perfil de tensão fase-neutro no primário 12,47 kV do ckt7
Figura 5.2	Distribuição das diferenças percentuais por fase no ckt7
Figura 5.3	Dispersão $V_{\text{programa}} \times V_{\text{DSS}}$
Figura 5.4	Curva de convergência do ckt7 ($\max \Delta S $ por iteração)
Figura 5.5	Magnitude da corrente de ramo no primário 12,47 kV do ckt7

LISTA DE TABELAS

Tabela	Título
Tabela 4.1	Origem dos componentes do programa: o que foi herdado e o que foi acrescentado
Tabela 5.1	Tensões em pu obtidas pelo programa no sistema-teste de 135 barras
Tabela 5.2	Características principais do circuito ckt7 do EPRI
Tabela 5.3	Diferenças percentuais de tensão no primário 12,47 kV do ckt7, por fase
Tabela 5.4	Diferenças percentuais de tensão no ckt7 estratificadas por nível
Tabela 5.5	Diferenças percentuais da corrente de ramo no ckt7
Tabela 5.6	Estatísticas globais de tensão para ckt5 e ckt24
Tabela 5.7	Diferenças medianas da corrente de ramo nos três circuitos
Tabela 5.8	Parâmetros do EnergySource declarados para cada circuito do EPRI

LISTA DE QUADROS

Quadro	Título
Quadro 2.1	Esqueleto da DFS recursiva para a varredura BFS
Quadro 3.1	Excerto da representação RDF/XML de um <code>ACLineSegment</code>
Quadro 3.2	Referência à biblioteca <code>TC57CIM.dll</code> no projeto <code>C#</code>
Quadro 4.1	Confinamento da <code>InvariantCulture</code> ao escopo do <code>getDSM</code>
Quadro 4.2	<i>Forward</i> adaptado para transformador
Quadro 4.3	Guarda anti- <i>NaN</i> para fases com tensão próxima de zero

LISTA DE ABREVIATURAS E SIGLAS

Sigla	Significado
ABNT	Associação Brasileira de Normas Técnicas
BFS	<i>Backward/Forward Sweep</i>
CIM	<i>Common Information Model</i>
CSV	<i>Comma-Separated Values</i>
DCIM	<i>Distribution Common Information Model</i>
DFS	<i>Depth-First Search</i>
DLL	<i>Dynamic-Link Library</i>
EMS	<i>Energy Management System</i>
EPRI	<i>Electric Power Research Institute</i>
FEIS	Faculdade de Engenharia de Ilha Solteira
IEC	<i>International Electrotechnical Commission</i>
LAPSEE	Laboratório de Planejamento de Sistemas de Energia Elétrica
LL	Linha-Linha (tensão fase-fase)
LN	Linha-Neutro (tensão fase-neutro)
OpenDSS	<i>Open Distribution System Simulator</i>
PF	<i>Power Flow</i>
PV	<i>Photovoltaic</i> / Barra de tensão controlada
RDF	<i>Resource Description Framework</i>
SDEE	Sistema de Distribuição de Energia Elétrica
SI	Sistema Internacional de Unidades
TC57	Comitê Técnico 57 do IEC (<i>Power systems management</i>)
TG	Trabalho de Graduação
UNESP	Universidade Estadual Paulista
XML	<i>eXtensible Markup Language</i>

LISTA DE SÍMBOLOS

Símbolo	Significado	Unidade
V_i	Vetor de tensões trifásicas na barra i	V (fase-neutro)
S_i	Vetor de potências aparentes da carga na barra i	VA
J_i	Vetor de correntes injetadas na barra i	A
I_i	Vetor de correntes complexas na barra i	A
Z_{abc}	Matriz 3×3 de impedância série em coordenadas de fase	Ω
Y_m	Matriz 3×3 de admitância <i>shunt</i> (não utilizada na varredura)	S
Z_s, Z_m	Impedância própria e mútua de uma linha trifásica equilibrada	Ω
Z_1, Z_0	Impedâncias de sequência positiva e zero	Ω
a	Relação de transformação U_1/U_2	—
U_1, U_2	Tensões nominais dos <i>ends</i> primário e secundário	V
S_{base}	Potência base do transformador	VA
R_{pu}, X_{pu}	Resistência e reatância em pu na base do trafo	pu
G, B	Condutância e susceptância <i>shunt</i> dos transformadores	S
ΔV	Resíduo de tensão entre iterações consecutivas	V
ΔS	Resíduo de potência	VA
tol_V	Tolerância de convergência em tensão ($= 10^{-3}$ V)	V
ε	Tolerância de convergência em potência ($= 10^{-4}$ pu)	pu

1 INTRODUÇÃO

1.1 CONTEXTO

O sistema de distribuição de energia elétrica passou, ao longo das duas últimas décadas, por uma transformação estrutural de larga escala. Em particular, a difusão da geração distribuída fotovoltaica, a inserção de cargas controláveis em pontos de consumo residencial e a obrigatoriedade regulatória de transparência operacional tornaram inviável a operação por meio de sistemas de informação compartimentados. Cada concessionária, em sua escala, gerencia hoje volumes de dados topológicos e elétricos da ordem de milhões de elementos, e a troca dessas informações com órgãos reguladores, fornecedores de equipamentos e simuladores externos passou a ser rotina (FERREIRA, 2021).

Nesse contexto, o desafio técnico desloca-se da execução do cálculo isolado para a interoperabilidade entre os sistemas. Um motor de fluxo de potência, por mais eficiente que seja, perde utilidade se cada par concessionária-fabricante exige formatos de arquivo distintos para descrever a mesma rede. A padronização da representação de sistemas elétricos é, dessa forma, condição necessária para a operação em escala industrial, e o esforço de normalização internacional concentrou-se em torno do *Common Information Model* (CIM), conforme apresentado no Capítulo 3.

A par dessa demanda de padronização, mantém-se atual o problema clássico do cálculo de fluxo de potência em alimentadores radiais. As particularidades da rede de distribuição (radialidade predominante, desequilíbrio entre fases, grande número de ramos curtos e presença de cargas monofásicas) dispensam o uso dos métodos de Newton-Raphson tradicionais e privilegiam algoritmos de varredura. O método *backward/forward* (BFS), apresentado em Kersting (2002) e formalizado em sua forma matricial 6×6 por Leite e Mantovani (2015), tornou-se referência nesse cenário, com implementações estabelecidas no grupo do Laboratório de Planejamento de Sistemas de Energia Elétrica (LAPSEE) da UNESP de Ilha Solteira (FERREIRA, 2021; SOUZA, 2023; GEROLOMETTO, 2023).

A confluência dos dois eixos, padronização CIM e fluxo de potência por varredura, sintetiza o escopo deste trabalho. Trata-se de fazer convergir, em um único aplicativo, a leitura de redes descritas em CIM/DCIM e o cálculo de fluxo de potência por BFS trifásica, com validação experimental frente ao simulador de referência do setor,

o OpenDSS (DUGAN; MONTENEGRO, 2020).

1.2 MOTIVAÇÃO

A motivação imediata deste trabalho deriva diretamente das conclusões de Ferreira (2021), o qual desenvolveu, no âmbito de seu Trabalho de Graduação, um motor de fluxo de potência trifásico para o sistema-teste de 135 barras do LAPSEE e um *parser* CIM para a leitura dos circuitos de distribuição do EPRI. As duas peças, no entanto, permaneceram desacopladas: o motor lia somente o XML proprietário de 135 barras, e o *parser* CIM, embora montasse em memória as estruturas correspondentes aos `TopologicalNode`, `ACLineSegment`, `PowerTransformer` e `EnergyConsumer` dos três circuitos do EPRI, não chegou a alimentar o cálculo de fluxo. Em palavras do próprio autor: *“em uma análise futura, o algoritmo de cálculo de fluxo de potência implementado deve ser utilizado para determinar as tensões e correntes nos três circuitos: ckt5, ckt7 e ckt24. Os valores obtidos devem ser comparados com aqueles obtidos usando o pacote de software do OpenDSS”* (FERREIRA, 2021, p. 65).

O presente trabalho fecha exatamente esse ciclo. A integração entre as duas camadas exige, primeiro, derivar dos atributos CIM brutos as grandezas eletricamente úteis (a matriz Z_{abc} a partir da `PerLengthSequenceImpedance`, o vetor S a partir do `EnergyConsumer`, os parâmetros físicos do transformador a partir das classes do pacote `AssetInfo`) e, em seguida, traduzir o modelo CIM para a hierarquia Rede/Barra/Ramo que alimenta o motor. A operação dos circuitos do EPRI introduz, ainda, duas dificuldades não enfrentadas em Ferreira (2021): a presença de transformadores em níveis intermediários do alimentador, exigindo modelagem física da relação de transformação na varredura; e a operação em grandezas físicas SI sem normalização em pu, o que invalida o critério de convergência tradicional baseado no resíduo das equações de potência.

Em paralelo, a escolha do CIM como padrão de entrada não é arbitrária. O perfil CIM 100 do EPRI, utilizado pelos três circuitos de teste, é a forma canônica de descrição adotada também pelo OpenDSS, fato que permite a comparação cruzada *bus-a-bus* sem necessidade de conversões intermediárias. Adicionalmente, a fundamentação matemática do BFS trifásico empregado neste trabalho coincide com a apresentada por Souza (2023), que trabalha sobre o mesmo formalismo de Leite e Mantovani (2015), propriedade que permite reaproveitar a hierarquia de classes e o

pseudocódigo da varredura sem alterações estruturais.

1.3 OBJETIVOS

1.3.1 Objetivo geral

Implementar e validar um algoritmo de fluxo de potência trifásico para sistemas de distribuição de energia elétrica modelados em CIM/DCIM, com base no perfil CIM 100 adotado pelo EPRI, e comparar os resultados obtidos com os do simulador OpenDSS em três circuitos de teste representativos: ckt5, ckt7 e ckt24.

1.3.2 Objetivos específicos

Em alinhamento ao objetivo geral, são objetivos específicos deste trabalho:

- (a) Estender o motor de fluxo de potência por varredura *backward/forward* desenvolvido por Ferreira (2021) e Souza (2023) para operar sobre redes descritas em CIM/DCIM, preservando a compatibilidade com o caminho original do sistema-teste de 135 barras.
- (b) Derivar a matriz de impedância 3×3 dos segmentos de linha a partir das instâncias `PerLengthSequenceImpedance` do CIM, conforme a formulação clássica de Kersting (2002).
- (c) Modelar os transformadores como ramos físicos da rede, incorporando a relação de transformação a na varredura, com tratamento unificado dos dois caminhos alternativos disponíveis no CIM 100 (`TransformerMeshImpedance` e `TransformerTank` com ensaios de curto-circuito e em vazio).
- (d) Construir o pipeline de leitura, tradução e cálculo em aplicativo único de console em C#, com geração automática dos relatórios elétricos formatados para o cruzamento *bus-a-bus* com a saída do OpenDSS.
- (e) Validar os resultados frente ao simulador OpenDSS em todos os três circuitos de teste do EPRI, com análise estatística dos desvios por barra e por fase, e identificação das fontes residuais de discrepância.

1.4 ORGANIZAÇÃO DO TRABALHO

O presente trabalho está organizado em seis capítulos. O Capítulo 2 apresenta a fundamentação do algoritmo de fluxo de potência trifásico por varredura, com a equação

matricial 6×6 que compacta as etapas *backward* e *forward*, a hierarquia de classes que materializa a árvore radial e o critério de convergência adotado. O Capítulo 3 descreve o modelo de informação comum (CIM/DCIM) em seu contexto normativo (IEC 61970-301 e IEC 61968-11), com ênfase nos pacotes Core, Topology, Wires e AssetInfo, e discute a estratégia de extensão por atributos privados que permite o cálculo de fluxo coabitar com a representação padronizada. O Capítulo 4 detalha a implementação em C#, com leitura do XML em CIM, correção do *locale*, extração de potência, cálculo da matriz Z_{abc} , modelagem dos dois caminhos de transformador no CIM, adaptação da DFS para incorporar a relação de transformação e tradução do modelo CIM para a hierarquia Rede. O Capítulo 5 apresenta os resultados obtidos no sistema-teste de 135 barras e a comparação detalhada com o OpenDSS no circuito ckt7, com análise estatística por fase, perfis de tensão, histograma das diferenças, dispersão entre os dois simuladores e discussão das fontes de desvio. Os circuitos ckt5 e ckt24 recebem tratamento sintético. O Capítulo 6 sintetiza as contribuições, registra as limitações conhecidas e enumera as direções de pesquisa futura. Ao final, são apresentadas as referências bibliográficas.

2 FLUXO DE POTÊNCIA TRIFÁSICO

Apresenta-se neste capítulo a fundamentação do algoritmo de fluxo de potência adotado no presente trabalho. A escolha recai sobre o método da varredura *backward/forward* (BFS) na sua versão trifásica em coordenadas de fase, formalizado por Leite e Mantovani (2015) e empregado em múltiplos trabalhos correlatos do grupo do LAP-SEE (FERREIRA, 2021; SOUZA, 2023). A apresentação privilegia a forma compacta da equação matricial 6×6 e a estrutura de dados que permite sua avaliação por busca em profundidade (*Depth-First Search*, DFS), preparando o terreno para as adaptações descritas no Capítulo 4.

2.1 ALGORITMO DE VARREDURA *BACKWARD/FORWARD*

O fluxo de potência em sistemas radiais de distribuição admite formulação particularmente eficiente quando se exploram duas propriedades estruturais: (i) a topologia em árvore, sem laços, garante caminho único da fonte a qualquer barra terminal; e (ii) a impedância série dos ramos é, em primeira aproximação, função apenas da geometria do condutor e do comprimento, e pode ser pré-calculada. Sob essas hipóteses, o problema reduz-se a duas operações encadeadas: a propagação das correntes a montante (a etapa *backward*) e a atualização das tensões a jusante (a etapa *forward*). A repetição alternada das duas operações leva à convergência das tensões ao perfil estacionário do alimentador.

Adotando a formulação de Leite e Mantovani (2015), a relação entre tensões e correntes em cada ramo do alimentador pode ser representada de forma compacta pela equação matricial 6×6 dada em (2.1):

$$\begin{bmatrix} [J_i]_{3 \times 1} \\ [V_i]_{3 \times 1} \end{bmatrix} = \begin{bmatrix} [A_i]_{3 \times 3} & [\bar{B}_i]_{3 \times 3} \\ [\bar{C}_i]_{3 \times 3} & [D_i]_{3 \times 3} \end{bmatrix} \begin{bmatrix} \sum_{d \in J(i)} [J_d]_{3 \times 1} + [(S_i/V_i)^*]_{3 \times 1} \\ [V_u]_{3 \times 1} \end{bmatrix} \quad (2.1)$$

em que V_i é o vetor de tensões fase-neutro na barra i , J_i é o vetor de correntes injetadas, S_i é a potência aparente trifásica da carga associada, V_u é a tensão da barra a montante (u de *upstream*), $J(i)$ é o conjunto de ramos a jusante de i , e as submatrizes A_i , $\bar{B}_i$, $\bar{C}_i$ e D_i dependem do tipo de ramo (linha, transformador, chave). Para o ramo de linha trifásica, $A_i = I_3$, $\bar{B}_i = 0$, $\bar{C}_i = -Z_{abc}$ e $D_i = I_3$, recuperando-se a forma clássica $V_i = V_u - Z_{abc} J_i$. Para o ramo de transformador, $\bar{C}_i$ e D_i incorporam a relação de transformação a e o deslocamento angular, conforme detalhado no Capítulo 4.

A equação (2.1) compacta, portanto, em uma única expressão, as duas etapas da varredura. A linha superior define o *backward*, em que a corrente injetada na barra i é a soma das correntes dos ramos a jusante mais a corrente da carga local, $(S_i/V_i)^*$. A linha inferior define o *forward*, em que a tensão na barra i é função da tensão a montante V_u e da queda no ramo. Souza (2023) e Ferreira (2021) reportam que a formulação se mostra estável e converge tipicamente em menos de dez iterações para alimentadores radiais bem condicionados, propriedade reconfirmada experimentalmente na seção 5.2.7 deste trabalho.

As correntes da carga $(S_i/V_i)^*$ introduzem uma não-linearidade no problema, dado que dependem das tensões, isto é, quantidades a determinar. Essa não-linearidade é tratada de forma iterativa: assume-se um perfil inicial de tensão (tipicamente $V_i = V_{u,0}$ em todas as barras), calcula-se a corrente da carga, executa-se a varredura completa, e atualiza-se o perfil. O laço externo termina quando o critério de convergência discutido na seção 2.4 é satisfeito.

2.2 REPRESENTAÇÃO COMPUTACIONAL DA REDE

A avaliação eficiente de (2.1) requer uma estrutura de dados que represente a árvore radial e permita percursos sistemáticos no sentido fonte-folha e folha-fonte. Adota-se, no presente trabalho, a hierarquia de classes herdada de Ferreira (2021) e mantida sem alteração estrutural, ilustrada na Figura 1.

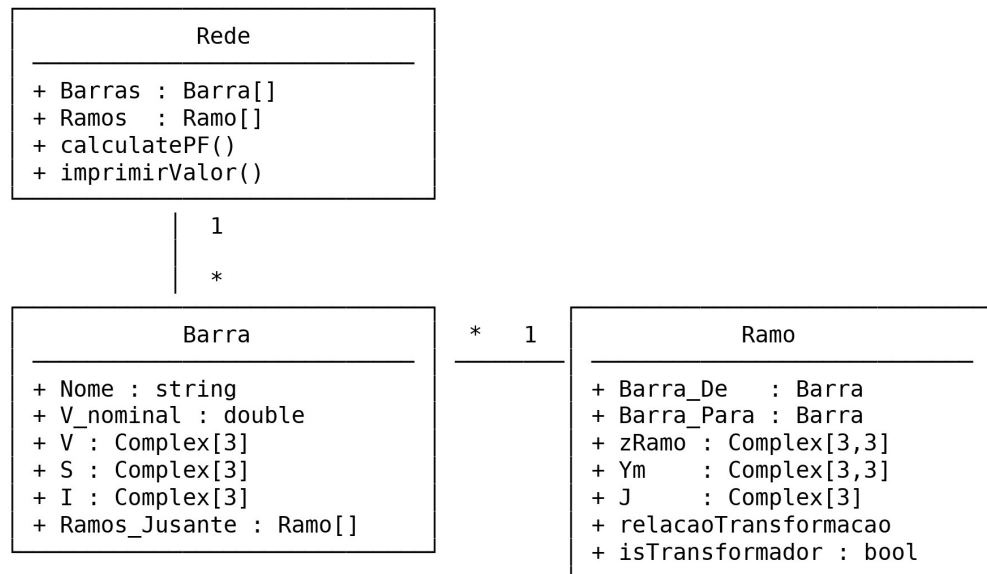


Figura 1 – Hierarquia de classes do modelo elétrico, com Rede agregando Barra[] e

Ramo[].

Fonte: Adaptado de Ferreira (2021).

A classe `Rede` opera como contêiner: agrega as listas de barras e ramos, mantém a ordem de varredura (definida pelo `processTopology` na etapa de orientação) e expõe os métodos públicos `calculatePF` e `imprimirValor`. A classe `Barra` representa cada nó topológico do alimentador e armazena o vetor de tensões complexas V por fase, a potência S da carga associada, a corrente I injetada e a referência aos ramos imediatamente a jusante, propriedade essencial para a recursão em profundidade. A classe `Ramo` representa a aresta orientada entre duas barras e armazena a matriz de impedância série Z_{abc} , a admitância *shunt* Y_m , o vetor de corrente J que percorre o ramo no sentido da varredura e, quando o ramo é um transformador, a relação de transformação a e a marca booleana `isTransformador`. Essa última distinção é a chave da adaptação descrita na seção 4.7.

Cumpra mencionar que a representação aqui adotada é deliberadamente compacta. Não há, por exemplo, classes auxiliares para representar reguladores de tensão ou bancos de capacitores chaveados; essas funcionalidades, presentes no OpenDSS, são tratadas neste trabalho como transformadores de relação fixa e como impedâncias *shunt* desligadas, respectivamente. A extensão da hierarquia para incorporar `RegulatingControl` é discutida no Capítulo 6 como direção de trabalho futuro.

De maneira análoga, a fonte do alimentador é representada por uma única `Barra` inicial cuja tensão é tomada como referência da varredura, sem a inclusão de um ramo de impedância equivalente que represente a rede a montante. O equivalente Thévenin do `EnergySource`, descrito no CIM por meio dos atributos `voltageMagnitude`, `voltageAngle`, `r`, `x`, `r0` e `x0` (seção 3.2), poderia ser incorporado à árvore radial como ramo fictício entre um nó interno e a barra `SOURCEBUS`. A presente implementação inicializa a raiz da varredura a partir do `nominalVoltage` da fonte, sem ler os atributos restantes do `EnergySource`. Essa escolha responde pelo pequeno resíduo de tensão de $\approx +1\%$ observado no primário na comparação com o OpenDSS (seção 5.2.8), retomado no Capítulo 6.

2.3 BUSCA EM PROFUNDIDADE RECURSIVA

A avaliação iterativa de (2.1) exige duas varreduras por iteração: uma no sentido folha-fonte para atualizar as correntes e outra no sentido fonte-folha para atualizar as tensões. Ambas podem ser efetuadas em um único percurso em profundidade da árvore,

conforme demonstrado por Souza (2023, fig. 4) e implementado de forma idêntica em Ferreira (2021). O Quadro 1 apresenta o pseudocódigo do percurso, em forma recursiva.

```

procedimento DFS(barra atual):
  para cada ramo r adjacente à barra atual:
    se r não foi visitado:
      marca r como visitado
      DFS(r.Barra_Para)                                (descida, visita filhos)
      backward:  $J[r] \leftarrow \sum J[r\_filho] + I[r.Para]$  (sobe corrente)
      forward :  $V[r.Para] \leftarrow V[r.De] - Z[r] \cdot J[r]$  (desce tensão)
    fim se
  fim para
fim procedimento

```

Quadro 1 – Esqueleto da DFS recursiva para a varredura *backward/forward*.

Fonte: Adaptado de Ferreira (2021).

O ponto sutil da formulação recursiva está na ordem das operações dentro do laço de ramos. O *backward* (soma das correntes) só pode ser executado **após** o retorno da chamada recursiva, pois requer as correntes já calculadas dos sub-ramos. O *forward* (queda de tensão), em contrapartida, deve ser executado **logo após** o *backward*, antes do retorno da função, para que a tensão atualizada na barra-filho esteja disponível à próxima descida. Essa ordem é preservada nas linhas 944–1023 do método `calculatePF` discutido no Capítulo 4.

A Figura 2 ilustra esquematicamente esse fluxo bidirecional na árvore radial.

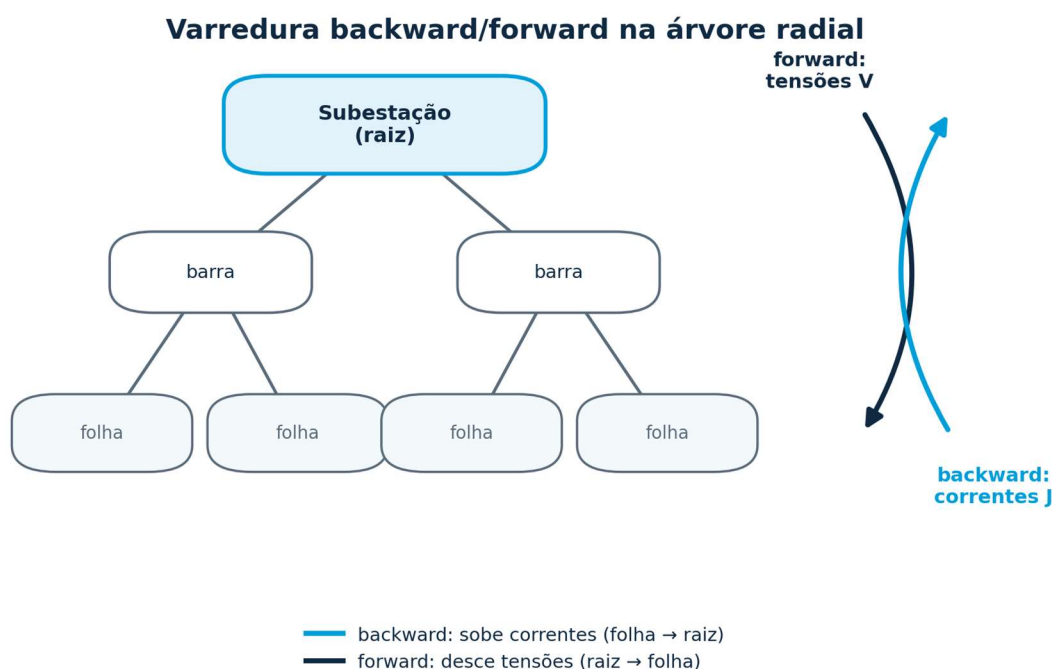


Figura 2 – Varredura *backward/forward* na árvore radial.

Fonte: Próprio autor.

A complexidade temporal do pseudocódigo do Quadro 1 é $O(N)$ por iteração, em que N é o número de ramos da árvore, propriedade que torna o método atraente para alimentadores de grande porte como o ckt7, com 1255 nós (seção 5.2). Souza (2023) demonstra que o mesmo pseudocódigo admite paralelização *fork-join*, com ganhos de até 33% em quatro núcleos; essa direção, porém, não é explorada no presente trabalho.

2.4 CRITÉRIO DE CONVERGÊNCIA

O laço externo da varredura termina quando as tensões da iteração corrente diferem das da iteração anterior por menos de uma tolerância pré-definida. A escolha da grandeza monitorada (e da unidade dessa grandeza) é o ponto sobre o qual incide a contribuição metodológica deste trabalho.

A formulação tradicional, adotada por Ferreira (2021), monitora o resíduo das equações de potência. Definindo $V_i^{(k)}$ como a tensão da barra i na k -ésima iteração e $\hat{S}_i^{(k)} = V_i^{(k)} (I_i^{(k)})^*$ como a potência aparente reconstruída a partir das grandezas calculadas, o resíduo de potência é:

$$\Delta S^{(k)} = \max_{i \in \mathcal{B}} |\hat{S}_i^{(k)} - S_i| \quad (2.2)$$

e o critério clássico é $\Delta S^{(k)} < \varepsilon$, com ε pequeno, tipicamente 10^{-4} pu para sistemas normalizados.

Para o sistema-teste de 135 barras, normalizado em pu, o critério (2.2) é adequado e converge tipicamente em menos de dez iterações. Para os circuitos do EPRI, no entanto, o programa opera em grandezas físicas SI (tensões em Volts, correntes em Ampères, potências ativa, reativa e complexa em Watts, VAR e VA, respectivamente): uma carga de 100 kW em estado quase-convergado com $\Delta V/V \approx 10^{-4}$ produz $|\Delta S| \approx 10$ VA, três ordens de grandeza acima de $\varepsilon = 10^{-4}$. O laço, nessas condições, estoura sistematicamente o limite de iterações sem que a tensão tenha de fato deixado de variar.

A solução adotada substitui o resíduo de potência pelo resíduo de tensão, conforme (2.3):

$$\Delta V^{(k)} = \max_{i \in \mathcal{B}} |V_i^{(k)} - V_i^{(k-1)}| \quad (2.3)$$

com critério de parada $\Delta V^{(k)} < \text{tol}_V$ e $\text{tol}_V = 10^{-3}$ V (1 mV). A escolha apresenta três vantagens. Primeiro, a grandeza monitorada tem interpretação física imediata: a

tolerância tol_V corresponde ao incremento mínimo perceptível no perfil de tensão. Segundo, a escala numérica é invariante quanto ao nível de tensão do circuito: uma tolerância de 1 mV é igualmente significativa em 240 V e em 12,47 kV, ao contrário de uma tolerância em pu, que requer base normalizada. Terceiro, em 12,47 kV linha-linha, 1 mV equivale a, aproximadamente, $1,4 \times 10^{-7}$ pu, dentro do limite numérico do tipo `double`, evitando falsos negativos por arredondamento.

A grandeza ΔS continua sendo gravada na variável `difs` ao longo das iterações, para fins de inspeção visual da convergência (Figura 8), mas perde a função de critério de parada. A escolha mantém, portanto, a interpretação física tradicional do balanço de potência como diagnóstico, ao mesmo tempo em que adota um critério numericamente estável para a terminação do laço.

3 MODELO DE INFORMAÇÃO COMUM (CIM/DCIM)

A interoperabilidade entre sistemas de informação no setor elétrico depende de uma representação padronizada dos elementos da rede. Este capítulo introduz o *Common Information Model* (CIM) na forma normatizada pelo IEC e descreve o subconjunto de pacotes mobilizados pelo presente trabalho. A apresentação privilegia os elementos efetivamente consumidos pelo programa em C# do Capítulo 4, sem ambição de exaustividade; leitores interessados nas demais áreas de aplicação podem consultar Ferreira (2021), que dedica capítulo inteiro ao tema.

3.1 DEFINIÇÃO E ABRANGÊNCIA

O CIM é um modelo conceitual unificado para a descrição de sistemas elétricos de potência, mantido pelo Comitê Técnico TC 57 da *International Electrotechnical Commission*. Sua origem remonta ao início da década de 1990, no âmbito da elaboração de uma interface comum entre sistemas de gerenciamento de energia (EMS, *Energy Management System*), e consolidou-se em duas normas complementares: a IEC 61970-301, voltada à operação em transmissão, e a IEC 61968-11, que estende o modelo para o domínio da distribuição. Conforme apresentado em IEC (2020) e em IEC (2013), o CIM define cerca de novecentas classes organizadas em pacotes funcionais, cada qual encapsulando um conjunto coeso de conceitos: geração, transmissão, ativos físicos, planejamento de manutenção, faturamento, entre outros.

Conforme apresentado por Ferreira (2021), a virtude central do CIM reside na separação entre o modelo conceitual e a forma de serialização. O modelo descreve apenas a semântica (quais elementos existem, quais relacionamentos os conectam, quais atributos cada classe carrega), deixando a codificação física como decisão de implementação. A serialização canônica adotada no presente trabalho é a forma RDF/XML, conforme será discutido na seção 3.4.

No contexto do CIM aplicado à distribuição (DCIM), o IEC 61968-11 adiciona classes específicas ao núcleo do CIM 61970, com destaque para o modelamento detalhado de transformadores de distribuição (*TransformerTank*, *TransformerTankInfo*), de cargas com discriminação por fase (*EnergyConsumerPhase*) e de elementos de proteção e manobra característicos da rede secundária. Os circuitos de teste do EPRI utilizados neste trabalho aderem ao perfil CIM 100, uma extensão privada que harmoniza os pacotes do IEC 61970 e do IEC 61968 com extensões de mercado norte-americano,

mantendo compatibilidade com a representação RDF/XML padrão.

3.2 PACOTES RELEVANTES PARA O FLUXO DE POTÊNCIA EM DISTRIBUIÇÃO

A modelagem completa de um alimentador no CIM mobiliza dezenas de classes, distribuídas em pacotes que cobrem aspectos físicos, topológicos e operacionais da rede. Para os fins deste trabalho, ou seja, a extração das grandezas eletricamente úteis ao cálculo de fluxo de potência, foram identificados quatro pacotes essenciais: **Core**, **Topology**, **Wires** e **AssetInfo**. A Figura 3 ilustra os relacionamentos entre as classes consumidas pelo parser desenvolvido.

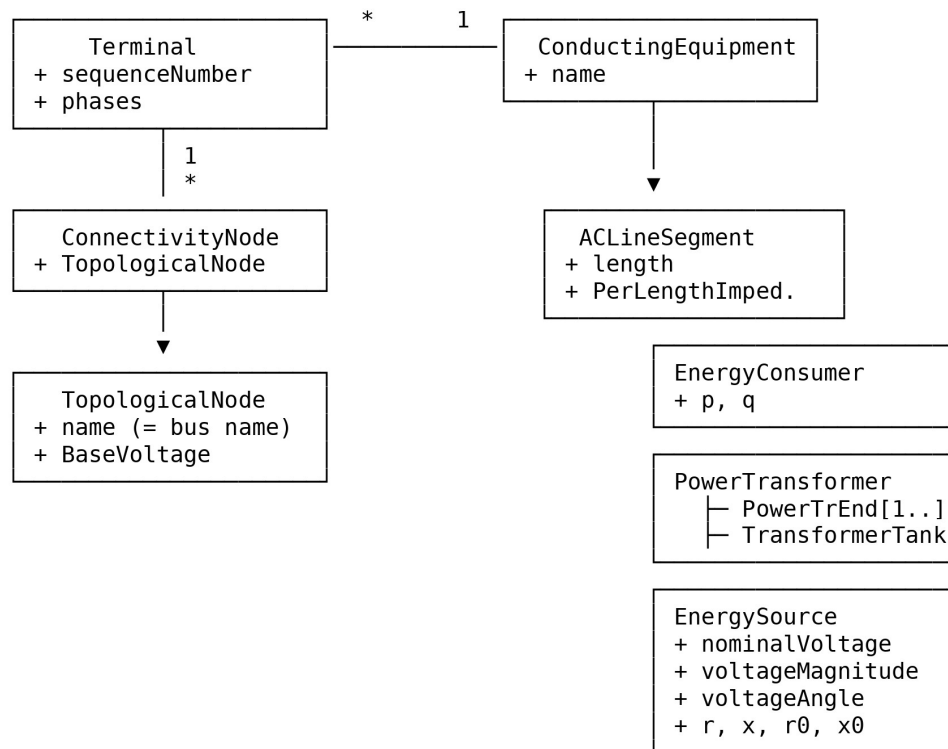


Figura 3 – Hierarquia simplificada das classes CIM consumidas pelo parser, sem distinção dos pacotes de origem. A classe `EnergySource` corresponde ao equivalente Thévenin da rede a montante.

Fonte: Adaptado de Ferreira (2021).

O **pacote Core** estabelece as classes fundamentais do CIM. A classe `IdentifiedObject` é a raiz da hierarquia e fornece os atributos `mRID` (identificador universal único) e `name` (rótulo legível). A classe `ConductingEquipment` é a abstração para qualquer elemento elétrico do circuito, e seus descendentes diretos abrangem desde geradores até cargas. A classe `Terminal` representa o ponto de conexão de um

ConductingEquipment a um ConnectivityNode, com atributos para a fase efetivamente conectada (phases) e a ordem do enrolamento no caso de transformadores (sequence-Number).

O **pacote Topology** introduz a noção de nó eletricamente único após o fechamento das chaves seccionadoras. A classe TopologicalNode agrupa, em tempo de análise, todos os ConnectivityNode interconectados por chaves fechadas, abstração diretamente correspondente à entidade Barra do programa. O atributo IdentifiedObject.name do TopologicalNode é, ainda, o elemento de costura entre o CIM e o relatório do OpenDSS, conforme discutido na seção 4.9.

O **pacote Wires** concentra as classes de equipamentos da rede. A ACLineSegment representa um trecho de linha trifásica e aponta para uma PerLengthSequenceImpedance que fornece r_1 , x_1 , r_0 e x_0 por unidade de comprimento. A classe EnergyConsumer modela a carga conectada a um TopologicalNode, com potência ativa e reativa totais; quando presente, a classe EnergyConsumerPhase discrimina a potência por fase. A classe PowerTransformer descreve o transformador e ramifica-se em duas vertentes: a primeira, mais antiga, agrega os PowerTransformerEnd (um por enrolamento) e a TransformerMeshImpedance (impedância de curto-circuito); a segunda, característica de transformadores de distribuição, agrega os TransformerTank e seus respectivos TransformerTankEnd, com a parametrização separada em uma TransformerTankInfo. Compõem ainda o pacote as classes Switch, Disconnecter, Fuse e Breaker, que representam dispositivos de manobra e proteção.

Integra o mesmo pacote a classe EnergySource, que representa o equivalente da rede a montante do ponto de entrada do alimentador. Seus atributos principais são nominalVoltage (tensão nominal da fonte, em Volts linha-linha), voltageMagnitude (tensão interna do equivalente Thévenin, em Volts), voltageAngle (ângulo de referência, em radianos), e o quarteto de impedâncias r , x , r_0 , x_0 (resistências e reatâncias de sequência positiva e zero do equivalente, em ohms). Os três circuitos de teste do EPRI declaram explicitamente todos esses atributos. No ckt7, por exemplo, o atributo voltageMagnitude vale 120,75 kV (razão exata de 1,05 sobre o nominalVoltage de 115 kV), e a impedância equivalente registra $r = 2,314 \Omega$ e $x = 28,52 \Omega$. A leitura completa desses atributos pelo parser, e a sua incorporação efetiva à árvore radial, são discutidas no Capítulo 4 (seção 4.8) e nos Capítulos 5 e 6.

O **pacote AssetInfo** complementa o pacote Wires com informações de fabricante e medições normalizadas dos ensaios de aceitação. A classe

TransformerTankInfo agrega os TransformerEndInfo (com ratedU, ratedS e connectionKind por enrolamento) e os resultados dos ensaios ShortCircuitTest (impedância de fuga e perdas em carga) e NoLoadTest (corrente de excitação e perdas em vazio). As fórmulas que convertem essas medições em parâmetros físicos do transformador estão apresentadas na seção 4.6.

3.3 EXTENSÃO DO CIM COM ATRIBUTOS PRIVADOS

O CIM, em sua concepção original, não foi pensado como ambiente de cálculo: ele descreve a topologia e os atributos físicos da rede, mas não armazena estados operacionais como o resultado de um fluxo de potência. Para que o cálculo possa coabitar com a representação CIM sem quebrar o padrão, adota-se a estratégia de extensão por atributos privados, conforme estabelecido por Ferreira (2021) e replicada no presente trabalho. A Figura 4 esquematiza a extensão proposta.

ACLineSegment (CIM 61970)		TopologicalNode (CIM Topology)	
+ length		+ name	
+ PerLengthSequenceImpedance		+ BaseVoltage	
+ Z [3][3] : Complex	] ext. priv.	+ S [3] : Complex	] ext. priv.
+ J [3] : Complex		+ I [3] : Complex	
		+ V [3] : Complex	

Figura 4 – Extensão do CIM com atributos privados de resultado: matriz Z e vetor J em ACLineSegment; vetores S , I e V em TopologicalNode.

Fonte: Adaptado de Ferreira (2021).

A extensão consiste em adicionar, na representação em memória das classes CIM, atributos privados de tipo Complex ou Complex[3] que carreguem o resultado do fluxo de potência. No ACLineSegment, anexa-se a matriz $Z_{3 \times 3}$ derivada da PerLengthSequenceImpedance pela equação (4.4) e o vetor $J_{3 \times 1}$ de correntes do ramo. No TopologicalNode, anexa-se o vetor $V_{3 \times 1}$ de tensões fase-neutro, o vetor $I_{3 \times 1}$ de correntes injetadas e o vetor $S_{3 \times 1}$ de potências aparentes da carga agregada.

Conforme demonstrado em Ferreira (2021), essa estratégia oferece dois benefícios. Em primeiro lugar, preserva a aderência ao padrão CIM: os atributos públicos da norma permanecem inalterados, e os atributos privados são invisíveis para sistemas externos que apenas leiam o XML serializado. Em segundo lugar, mantém a localidade da informação: o resultado do cálculo para uma dada barra fica armazenado no próprio TopologicalNode, dispensando estruturas de dicionários externos que

precisariam ser sincronizadas com a topologia. No programa em C#, essa extensão materializa-se nas classes-espelho `DadosZJRamoCimACLine`, `DadosSCimCarga` e `DadosTrafoCim`, que vivem na camada CIM (CPDSM) e são consumidas pela tradução para o modelo Barra/Ramo.

Cumprе mencionar que a literatura recente reconhece duas alternativas a essa estratégia. A primeira consiste em definir um perfil CIM dedicado a resultados de fluxo de potência (CIM-PF), com classes próprias para `BusValue`, `BranchFlow` e similares; essa alternativa, embora aderente ao espírito do padrão, exigiria a publicação de um perfil próprio e a recompilação da biblioteca `TC57CIM.dll` mencionada na seção 3.4. A segunda consiste em armazenar os resultados em uma base de dados relacional externa, indexada pelos `mRID` das classes CIM, abordagem detalhada por Barros e Leite (2021) no contexto de redes de distribuição. A extensão por atributos privados adotada aqui equilibra simplicidade de implementação e fidelidade ao modelo conceitual, e mostrou-se suficiente para os fins do trabalho.

3.4 SERIALIZAÇÃO RDF/XML E A BIBLIOTECA TC57CIM.DLL

A serialização canônica das instâncias CIM segue o formato RDF/XML, normatizado pela W3C e adotado pelo IEC TC 57 como representação de referência. Cada instância de uma classe CIM corresponde a um elemento XML cujo nome resolve-se via *namespace* `cim`, e seus atributos primitivos são apresentados como elementos filhos. Os relacionamentos com outras instâncias são representados por elementos cujo atributo `rdf:resource` aponta para o `rdf:ID` da instância alvo. A título de ilustração, o Quadro 2 reproduz um trecho do XML do circuito `ckt7`, com a descrição de um `ACLineSegment` típico.

```
<cim:ACLineSegment rdf:ID="_2AB3F1F0-3F7F-...">
  <cim:IdentifiedObject.name>HARMSEM-21013</cim:IdentifiedObject.name>
  <cim:ConductingEquipment.BaseVoltage rdf:resource="#_BV_12_47kV"/>
  <cim:Conductor.length>304.78</cim:Conductor.length>
  <cim:ACLineSegment.PerLengthImpedance
    rdf:resource="#_PerL_OH_4ACSR_2_4ACSR_2_4ACSR"/>
  <cim:Equipment.EquipmentContainer rdf:resource="#_LINE_CONTAINER_03"/>
</cim:ACLineSegment>
```

Quadro 2 – Excerto da representação RDF/XML de um `ACLineSegment` no `ckt7`. Os atributos `rdf:resource` referenciam outras instâncias do mesmo arquivo.

Fonte: Próprio autor.

A leitura programática de arquivos com essa estrutura exigiria, em princípio, a implementação manual de cada classe CIM no projeto, com seus atributos e

relacionamentos. Para evitar esse esforço, que envolveria a tradução de aproximadamente novecentas classes da norma, optou-se por reaproveitar a biblioteca **TC57CIM.dll**, compilada por Ferreira (2021) a partir do projeto de referência do IEC TC 57. A biblioteca expõe os pacotes Core, Topology, Wires e AssetInfo em espaços de nomes coerentes com a estrutura da norma (IEC61970.Base.Core, IEC61970.Base.Topology, IEC61970.Base.Wires, IEC61968.AssetInfo), com cada classe CIM mapeada para uma classe C# correspondente.

A inclusão da biblioteca no projeto é feita por referência binária no arquivo `consoleApp.csproj`, conforme reproduzido no Quadro 3.

```
<Reference Include="TC57CIM">  
  <HintPath>..\dll\TC57CIM.dll</HintPath>  
</Reference>
```

Quadro 3 – Referência à biblioteca TC57CIM.dll no projeto C#.

Fonte: Próprio autor.

A versão da TC57CIM.dll empregada cobre o perfil CIM 100, que é o adotado pelo EPRI nos três circuitos de teste utilizados. Eventuais ampliações futuras da norma (CIM 101, por exemplo) podem ser absorvidas pela substituição da DLL, sem reescrita do parser do Capítulo 4, propriedade que reforça o desacoplamento entre a camada de leitura e a camada de cálculo, conforme já discutido na seção 2.2.

Por fim, registra-se que o arquivo XML do CIM 100 não impõe ordenação dos elementos nem garante coerência referencial inteira: ocorrem, na prática, atributos `rdf:resource` que apontam para instâncias declaradas em arquivos auxiliares (`ckt7_CIM100x_DI.XML` para o *display schema*, por exemplo). O parser desenvolvido neste trabalho opera apenas sobre o arquivo principal, fazendo uso da tolerância a referências não resolvidas característica da biblioteca TC57CIM.dll: instâncias ausentes geram referência nula no objeto correspondente, permitindo que a leitura prossiga e que os atributos primários (único interesse do cálculo elétrico) sejam acessados sem erro.

4 METODOLOGIA: IMPLEMENTAÇÃO EM C#

Apresenta-se neste capítulo a implementação computacional desenvolvida em linguagem C# que estende o motor de fluxo de potência herdado de Ferreira (2021) e Souza (2023) para operar diretamente sobre redes descritas em CIM/DCIM, conforme especificadas nos circuitos de teste do EPRI. As escolhas de projeto seguem a hierarquia exposta no Capítulo 2 e o conjunto de pacotes CIM relevantes apresentado no Capítulo 3.

4.1 VISÃO GERAL DA ARQUITETURA

A solução foi construída em projeto único de aplicativo de console, denominado `ConsoleApp`, organizado em dois espaços de nomes que refletem as duas camadas de abstração do problema. O pacote `modeloRD` reúne as estruturas elétricas (classes `Rede`, `Barra` e `Ramo`) e concentra o algoritmo de varredura. O pacote `CPDSM` é responsável pela leitura do XML em CIM e pela montagem das estruturas intermediárias que servem de ponte entre os dois mundos. A separação entre as duas camadas obedece à diretriz adotada por Ferreira (2021): manter o cálculo elétrico independente do formato de entrada, permitindo a substituição da fonte de dados sem reescrita do motor de fluxo.

A Figura 4.1 ilustra o pipeline de execução, cuja entrada principal é o método `Main` do projeto (`Program.cs:L2727`). Após a seleção do circuito pelo menu interativo, no item correspondente à base do EPRI, o fluxo percorre cinco etapas encadeadas: (i) leitura do arquivo XML pelo método `getDSM`, que popula em memória os arrays correspondentes a cada classe CIM utilizada; (ii) descoberta topológica pelo método `processTopology`, que percorre os `Terminal` e identifica nós ascendentes e descendentes do alimentador via busca em profundidade; (iii) tradução do modelo CIM para o modelo elétrico pela rotina `converterCimParaRede`, que instancia as classes `Barra` e `Ramo` a partir dos `TopologicalNode`, `ACLineSegment` e `PowerTransformer`; (iv) execução do fluxo de potência pelo método `calculatePF`, com a varredura *backward/forward* recursiva adaptada para incorporar a relação de transformação dos transformadores; e (v) escrita dos quatro relatórios em disco, formatados para o cruzamento *bus-a-bus* com a saída do OpenDSS.

Adicionalmente à leitura do CIM, o programa preserva o caminho de execução voltado ao sistema-teste de 135 barras de domínio do LAPSEE, o qual emprega um XML proprietário e foi originalmente desenvolvido por Ferreira (2021). A coexistência

dos dois caminhos no mesmo executável permite verificar regressões de comportamento sempre que o motor de fluxo é modificado, conforme se discute na seção 5.1.

A Tabela 1 consolida a origem de cada componente do programa, com a distinção entre o que foi herdado do código de Ferreira (2021) e o que foi acrescentado neste trabalho. A apresentação serve de roteiro para as seções subsequentes do capítulo, que detalham individualmente cada bloco acrescentado.

Componente	Origem	Forma na presente implementação
Hierarquia Rede, Barra e Ramo	Ferreira (2021)	Preservada sem alteração estrutural; campos Nome, V_nominal, isTransformador e relacaoTransformacao ampliados
Motor BFS recursivo (calculatePF)	Ferreira (2021)	Estendido para tratar o transformador como ramo físico, com divisão pela relação a no <i>forward</i> e no <i>backward</i> , e protegido por guarda anti- <i>NaN</i>
Parser do XML proprietário do LAPSEE	Ferreira (2021)	Mantido sem alteração, em coexistência com o parser CIM
Parser CIM RDF/XML (getDSM)	Ferreira (2021)	Estendido para TransformerTank, TransformerTankInfo, TransformerEndInfo, ShortCircuitTest e NoLoadTest. Confinamento da CultureInfo.InvariantCulture ao escopo do método
Cardinalidade de TransformerCoreAdmittance e TransformerMeshImpedance	Ferreira (2021)	Convertida de instância única para arrays, com dicionários por mRID para acesso em $O(1)$
Descoberta topológica (processTopology)	Ferreira (2021)	Preservada; reaproveitada pela orientação dos ramos na tradução CIM para Rede
Critério de convergência	Ferreira (2021)	Substituído de $\ \Delta S\ < \varepsilon$ para $\max \ \Delta V\ < tol_V$

Componente	Origem	Forma na presente implementação
Tratamento do laço externo sem EnergySchedulingType.PV	Ferreira (2021)	Inicialização explícita de $erro_max_gd = 0$ quando o vetor de geradores é vazio, condição essencial para os circuitos do EPRI
Extração de potência por barra (MontarDicionariosPotenciaCargaCim)	Acrescentado neste trabalho	Cobre os dois caminhos do CIM (com e sem EnergyConsumerPhase), com agregação por TopologicalNode
Cálculo da matriz Z_{abc} 3×3 (PreencherZRamoEJaPartirSequenciaPerLength)	Acrescentado neste trabalho	Recompõe Z_{abc} a partir de Z_1 e Z_0 da PerLengthSequenceImpedance
Modelagem de transformadores (MontarDicionarioTransformadoresCim, ConstruirTrafoCaminhoA, ConstruirTrafoCaminhoB)	Acrescentado neste trabalho	Cobre os dois caminhos do CIM 100 (caminho A com TransformerMeshImpedance, caminho B com TransformerTank mais ensaios)
Classes-espelho DadosZJRamoCimACLine, DadosSCimCarga, DadosTrafoCim	Acrescentado neste trabalho	Contêineres de dados eletricamente úteis derivados do CIM, indexados por mRID, descritos na seção 4.8
Tradução CIM para Rede (converterCimParaRede)	Acrescentado neste trabalho	Instancia Barra e Ramo a partir dos dicionários, com orientação por DFS espelhando processTopology e propagação de $V_{nominal}$ por BFS
Quatro relatórios em texto (resultadoPF_*, MatricesZ_*, PotenciasCarga_*, Transformadores_*)	Acrescentado neste trabalho	Formato compatível com cruzamento bus-a-bus por mRID e por Nome (TopologicalNode.IdentifiedObject.name) com a saída do OpenDSS

Tabela 1 – Origem dos componentes do programa: o que foi herdado de Ferreira (2021) e o que foi acrescentado na presente implementação.

Fonte: Próprio autor.

4.2 A BIBLIOTECA TC57CIM.DLL

A leitura das instâncias do CIM exigiria, em princípio, a implementação completa dos pacotes IEC 61970 e IEC 61968 no projeto. Para evitar essa redundância, optou-se por reaproveitar a biblioteca TC57CIM.dll, compilada por Ferreira (2021) a partir do projeto de referência do IEC TC57. A biblioteca expõe, nos espaços de nomes IEC61968.AssetInfo e IEC61970.Base.*, as classes geradas que correspondem a cada classe CIM da norma, com seus atributos e relacionamentos preservados. A inclusão no projeto é feita por referência binária no arquivo ConsoleApp.csproj (linhas 38–42), e a versão utilizada cobre o perfil CIM 100, adotado pelo EPRI nos circuitos ckt5_CIM100x.XML, ckt7_CIM100x.XML e ckt24_CIM100x.XML.

A escolha da biblioteca traz duas consequências relevantes. Em primeiro lugar, o parser desenvolvido neste trabalho não constrói diretamente as instâncias CIM: ele apenas mapeia os elementos do XML para as classes já existentes na DLL, usando reflexão e o `XmlNamespaceManager` para resolver os prefixos `cim`, `entsoe` e `rdf`. Em segundo lugar, eventuais ampliações futuras da norma podem ser absorvidas pela substituição da DLL, sem reescrita do parser, propriedade discutida no Capítulo 6.

4.3 PARSER CIM E CORREÇÃO DO LOCALE

O método `getDSM` da classe `CPDSM` (Program.cs:L1415–L1437) é o ponto de entrada da leitura do XML. Para cada classe CIM relevante, define-se um *XPath* compatível com o `XmlNamespaceManager` e percorrem-se os nós retornados, populando arrays tipados sobre a biblioteca TC57CIM. A estratégia segue a abordagem original de Ferreira (2021), aqui estendida com a leitura das classes `TransformerTank`, `TransformerTankInfo`, `TransformerEndInfo`, `ShortCircuitTest` e `NoLoadTest`, essenciais para o caminho B de modelagem dos transformadores apresentado na seção 4.6.

Durante a verificação inicial do código herdado, observou-se que o circuito ckt7 do EPRI divergia logo nas primeiras iterações de fluxo, com valores de impedância da ordem de $10^9 \Omega$. Investigação detalhada revelou que o atributo `cim:TransformerEndInfo.ratedU`, com valor original 7199.5616 V, era interpretado como 71995616 V durante a conversão para `double`. A causa raiz reside no comportamento do método `convert.ToSingle` em ambiente *pt-BR*, no qual o ponto é tratado como separador de milhares e a vírgula como separador decimal: a sequência 7199.5616 é interpretada como um único inteiro 71995616. Como o cálculo subsequente de $R_{\Omega} = R_{pu} U_1^2 / S_{base}$ é

quadrático em U_1 , o erro propagou-se em 10^8 , tornando a varredura instável.

A correção consistiu em forçar `CultureInfo.InvariantCulture` no escopo da leitura, conforme reproduzido no Quadro 4.

```
CultureInfo culturaOriginal = Thread.CurrentThread.CurrentCulture;
Thread.CurrentThread.CurrentCulture = CultureInfo.InvariantCulture;
try { /* todo o parsing CIM */ }
finally { Thread.CurrentThread.CurrentCulture = culturaOriginal; }
```

Quadro 4 – Confinamento da cultura *InvariantCulture* ao escopo do `getDSM`.

Fonte: Próprio autor.

A restauração da cultura original no bloco `finally` foi adotada por uma razão técnica relevante. O caminho do sistema-teste de 135 barras, herdado de Ferreira (2021), utiliza vírgula como separador decimal no XML proprietário do LAPSEE, escolha não-padrão segundo a especificação XML, porém funcional sob *locale pt-BR*. Uma alteração global da cultura comprometeria a retrocompatibilidade do programa com aquele caminho. O confinamento da `InvariantCulture` ao escopo de `getDSM` preserva, portanto, ambas as bases de teste.

Adicionalmente, a leitura do CIM foi tornada mais confiável no que diz respeito à cardinalidade de algumas classes. As propriedades `TransformerCoreAdmittance` e `TransformerMeshImpedance`, declaradas como singulares no código herdado, foram convertidas em arrays na presente implementação. A motivação é quantitativa e foi identificada na primeira execução sobre os circuitos do EPRI: o `ckt7_CIM100x.XML` declara mais de uma dezena de instâncias de cada uma dessas classes, e o `ckt24_CIM100x.XML` declara várias centenas, das quais o código herdado lia apenas a primeira ocorrência. O efeito silencioso era a aplicação de um único modelo de impedância de curto-circuito a todos os transformadores do alimentador, sem mensagem de erro em tempo de execução. Os dicionários `corePorEndMrid` e `meshPorFromEndMrid`, introduzidos em `MontarDicionarioTransformadoresCim`, permitem o acesso em $O(1)$ ao mRID do `PowerTransformerEnd` correspondente, evitando varreduras quadráticas na etapa de tradução.

4.4 EXTRAÇÃO DE POTÊNCIA POR BARRA

O fluxo de potência radial pressupõe o conhecimento prévio das injeções de potência aparente S_i em cada barra do alimentador. No CIM, essa informação é distribuída entre as instâncias da classe `EnergyConsumer` e, opcionalmente, de seus `EnergyConsumerPhase`. O método `MontarDicionariosPotenciaCargaCim` (`Program.cs:L2368–L2440`) consolida essa informação em dois dicionários: `cargaSPorEnergyConsumerMrid`, indexado

pelo identificador único do consumidor, e noSPorTopologicalNodeMrid, agregado por nó topológico, esse último utilizado diretamente pela tradução para Barra.S.

A construção dos dicionários percorre os EnergyConsumer e ramifica-se em dois caminhos. No primeiro caminho, quando o consumidor possui filhos da classe EnergyConsumerPhase, cada fase carrega seus próprios valores de potência ativa P_ϕ e reativa Q_ϕ . A potência complexa atribuída à fase $\phi \in \{a, b, c\}$ é, portanto, $S_\phi = P_\phi + jQ_\phi$. No segundo caminho, mais comum nos circuitos do EPRI, o EnergyConsumer registra apenas a potência total trifásica em seus atributos p e q. Adota-se, neste caso, a distribuição uniforme expressa em (4.1):

$$S_a = S_b = S_c = \frac{P_{\text{total}} + jQ_{\text{total}}}{3} \quad (4.1)$$

A escolha pela divisão equitativa, em vez de qualquer ponderação por fase, decorre da ausência de informação adicional no CIM 100 que pudesse justificar tratamento assimétrico. Como se discute na seção 5.2.8, parte dos desvios observados em ramos com cargas monofásicas e bifásicas pode ser parcialmente atribuída a essa simplificação, embora seu impacto na tensão da barra-fonte do alimentador seja desprezível.

A consolidação por nó topológico (realizada no segundo passo do método) soma as contribuições de todos os EnergyConsumer conectados ao mesmo TopologicalNode. Essa agregação reflete a natureza do TopologicalNode no CIM, que representa um nó eletricamente único após o fechamento das chaves seccionadoras. O dicionário resultante alimenta a inicialização de Barra.S na etapa de tradução, apresentada na seção 4.8.

4.5 CÁLCULO DA MATRIZ DE IMPEDÂNCIA 3×3 DAS LINHAS

Os segmentos de linha do CIM são descritos pela classe ACLineSegment, que aponta para uma PerLengthSequenceImpedance contendo as impedâncias de sequência positiva (r_1, x_1) e zero (r_0, x_0) por unidade de comprimento. O método PreencherZRamoeJPartirSequenciaPerLength (Program.cs:L2441–L2493) converte essa representação na matriz de impedância série 3×3 em coordenadas de fase, forma necessária à varredura no domínio das fases.

Inicialmente, multiplicam-se os parâmetros por unidade de comprimento pelo length do ACLineSegment, conforme (4.2):

$$Z_1 = (r_1 + jx_1)L \quad Z_0 = (r_0 + jx_0)L \quad (4.2)$$

Em seguida, recuperam-se as impedâncias próprias (Z_s) e mútuas (Z_m) pela

transformação inversa da decomposição de Fortescue para a linha trifásica equilibrada em construção (KERSTING, 2002):

$$Z_s = \frac{Z_0 + 2Z_1}{3} \quad Z_m = \frac{Z_0 - Z_1}{3} \quad (4.3)$$

A matriz de impedância Z_{abc} resulta da montagem cíclica das grandezas de (4.3):

$$Z_{abc} = \begin{bmatrix} Z_s & Z_m & Z_m \\ Z_m & Z_s & Z_m \\ Z_m & Z_m & Z_s \end{bmatrix} \quad (4.4)$$

A matriz dada em (4.4) é a representação adequada para alimentadores radiais com linhas curtas, nos quais o efeito *shunt* Y_m pode ser desconsiderado em primeira aproximação. No código, mantém-se o campo `Ramo.Ym` apto a receber a admitância mútua, porém com cálculo desabilitado em `calculatePF` durante a varredura, por motivos de estabilidade numérica (Program.cs:L941). A reativação do termo *shunt* é discutida no Capítulo 6.

Cumprе mencionar que a parametrização de (4.3) pressupõe simetria estrutural da linha, hipótese válida para configurações de cabo coberto e linhas aéreas com transposição completa. Configurações assimétricas (cabos paralelos com afastamento desigual ou linhas sem transposição) exigiriam o emprego das equações de Carson, conforme tratamento detalhado por Kersting (2002). Esses casos não estão representados nos circuitos `ckt5`, `ckt7` e `ckt24` com nível de detalhe suficiente, razão pela qual o programa opta pela formulação compacta de (4.4).

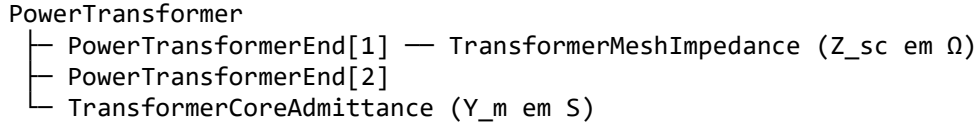
A matriz Z_{abc} é armazenada no campo `zRamo` da classe `Ramo` e consumida pela varredura na expressão (4.9) da seção 4.7. Concomitantemente, o vetor de correntes $J = [J_a, J_b, J_c]^T$ do ramo é inicializado em zero, sendo preenchido em tempo de execução durante a varredura *backward*.

4.6 MODELAGEM DE TRANSFORMADORES COMO RAMO FÍSICO

A representação dos transformadores constitui a contribuição de maior peso técnico deste trabalho. O CIM 100, em sua versão consolidada pelo EPRI, oferece duas formas alternativas (e mutuamente exclusivas) para a descrição de transformadores em redes de distribuição. O método `MontarDicionarioTransformadoresCim` (Program.cs:L2031–L2148) identifica automaticamente qual dos dois caminhos foi utilizado em cada `PowerTransformer` e delega a montagem dos parâmetros à rotina apropriada: `ConstruirTrafoCaminhoA` OU `ConstruirTrafoCaminhoB`.

4.6.1 Caminho A: TransformerMeshImpedance

O primeiro caminho, predominante em transformadores de subestação descritos no ckt5, segue a estrutura clássica do IEC 61970, com a impedância e a admitância já fornecidas em valores físicos absolutos. A relação de classes é a seguinte:



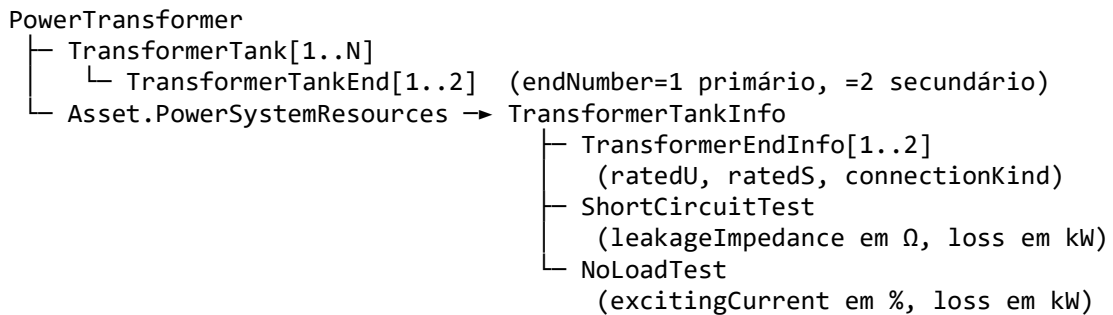
Nesse caminho, os atributos `ratedU` e `ratedS` dos `PowerTransformerEnd` fornecem diretamente as tensões nominais por enrolamento e a potência base, ao passo que `TransformerMeshImpedance` contém a impedância de curto-circuito referida ao *end* energizado. A matriz Z_{trafo} é instanciada como diagonal $\text{diag}(Z_{sc}, Z_{sc}, Z_{sc})$ e projetada sobre as fases conectadas, conforme o atributo `phases` do `PowerTransformerEnd`. A relação de transformação é dada por (4.5):

$$a = \frac{U_1}{U_2} \quad (4.5)$$

em que U_1 e U_2 correspondem aos `ratedU` do *end* primário e do *end* secundário, respectivamente.

4.6.2 Caminho B: TransformerTank com ensaios

O segundo caminho, característico de transformadores de distribuição descritos no ckt7 e no ckt24, parte de um modelo de ativo (`Asset`) que abriga as medições normalizadas dos ensaios de curto-circuito e de excitação:



Diferentemente do caminho A, a impedância de curto-circuito não vem em forma fechada: ela é derivada das medições do ensaio de curto-circuito, expressa primeiro em pu na base própria do trafo e em seguida convertida para ohms na base do *end* primário. A sequência de transformações, aplicada em `ConstruirTrafoCaminhoB` (`Program.cs:L2238–L2367`), está reproduzida nas equações (4.6) a (4.8):

$$R_{pu} = \frac{P_{loss} [\text{kW}] \cdot 10^3}{S_{base}} \quad Z_{pu} = \frac{Z_{leak} [\Omega] \cdot S_{base}}{U_{test}^2} \quad X_{pu}$$

$$= \sqrt{\max(0, Z_{pu}^2 - R_{pu}^2)} \quad (4.6)$$

$$R_{\Omega} = R_{pu} \frac{U_1^2}{S_{base}} \quad X_{\Omega} = X_{pu} \frac{U_1^2}{S_{base}} \quad (4.7)$$

De maneira análoga, o ensaio em vazio é convertido em admitância *shunt*, conforme (4.8):

$$G_{pu} = \frac{P_{noLoad} [\text{kW}] \cdot 10^3}{S_{base}} \quad Y_{pu} = \frac{I_{ex} [\%]}{100} \quad B_{pu} = \sqrt{\max(0, Y_{pu}^2 - G_{pu}^2)} \quad (4.8)$$

A matriz Y_m resultante é armazenada no campo `Ramo.Ym` para uso futuro, embora não seja utilizada na varredura por motivos de estabilidade, decisão análoga à tomada em (4.4) para o termo *shunt* das linhas. A subtração contida na raiz quadrada de (4.6) e (4.8) é protegida pela função $\max(0, \cdot)$ para tolerar imprecisões dos ensaios reportados no CIM, situação em que a soma das parcelas R^2 e G^2 pode exceder marginalmente Z^2 e Y^2 por arredondamento.

A confiabilidade da rotina diante das variantes encontradas nos arquivos do EPRI foi conferida ramo a ramo. Em particular, registra-se a presença de transformadores monofásicos em bancos de três, situação na qual cada `TransformerTank` representa um único trafo de fase e a consolidação em Z_{abc} é feita por composição. A função `ConstruirTrafoCaminhoB` lida com esse arranjo identificando o atributo `phase` em cada `TransformerTankEnd` e atribuindo o Z à fase correspondente, com zeros nas demais posições.

4.7 ADAPTAÇÃO DA DFS PARA INCORPORAR TRANSFORMADORES

A varredura *backward/forward*, conforme apresentada no Capítulo 2, opera sobre ramos do tipo linha, nos quais a queda de tensão é dada por $V_{para} = V_{de} - ZJ$ no sentido *forward*, e a corrente de cada ramo é a soma das correntes dos ramos a jusante no sentido *backward*. A inclusão do transformador como ramo físico, não como simples impedância equivalente, exige duas adaptações nessas duas operações, formuladas neste trabalho e implementadas em `calculatePF` (Program.cs:L944–L1023).

No sentido *forward*, a tensão na barra secundária do transformador é obtida pela queda na impedância de curto-circuito Z_{trafo} referida ao primário, seguida da reflexão para o nível secundário pela relação de transformação a :

$$V_{para} = \frac{V_{de} - Z_{trafo} J_{pri}}{a} \quad (4.9)$$

No sentido *backward*, a corrente que chega ao secundário é refletida ao primário pela conservação dos Ampère-espiras no transformador ideal:

$$J_{pri} = \frac{J_{sec}}{a} \quad (4.10)$$

Para ramos do tipo linha, adota-se $a = 1$, e as expressões (4.9) e (4.10) recaem nas formas clássicas. Essa unificação permite que o pseudocódigo da varredura permaneça único: o tratamento condicional fica restrito à divisão por a ao final do laço de fase, conforme reproduzido no Quadro 5.

```
for (int i = 0; i < 3; i++)
{
    Barra_Para.V[i] = Barra_De.V[i];
    for (int j = 0; j < 3; j++)
        Barra_Para.V[i] -= zRamo[i][j] * J[j];

    if (isTransformador && relacaoTransformacao > 0)
        Barra_Para.V[i] /= relacaoTransformacao;
}
```

Quadro 5 – Forward adaptado para transformador. A divisão por a refere a queda na impedância de dispersão ao secundário.

Fonte: Próprio autor.

A formulação adotada equivale à transferência das grandezas para o secundário pela substituição $Z_{sec} = Z_{pri}/a^2$ e $J_{sec} = a J_{pri}$, escolha que evita a manutenção de uma segunda matriz Z_{sec} ao custo de uma divisão complexa por fase. A equivalência foi verificada em rede de teste mínima de duas barras conectadas por um transformador ideal de relação fixa, com erro inferior a 10^{-9} V em relação à solução analítica.

Adicionalmente à reflexão da relação de transformação, foi necessário introduzir uma guarda numérica para situações em que a magnitude da tensão complexa em uma fase se aproxima do limite numérico do tipo `double`. Em transitórios mal condicionados (carga monofásica em fase sem retorno por neutro, por exemplo) o cálculo direto $I = (S/V)^*$ produziria *NaN*. A guarda, implementada em `Program.cs:L981–L995`, zera a corrente nesse caso:

```
const double V_FLOOR = 1e-9;
if (Vp.Magnitude > V_FLOOR)
    Barra_Para.I[i] = Complex.Conjugate(Sbus / Vp);
else
    Barra_Para.I[i] = Complex.Zero;
```

Quadro 6 – Guarda anti-*NaN* para fases com tensão próxima de zero.

Fonte: Próprio autor.

Por fim, a substituição do critério de convergência (de $|\Delta S| < \varepsilon$ para $\max|\Delta V| < \text{tol}_V$, conforme exposto na seção 2.4) mostrou-se indispensável para que o laço terminasse em redes não normalizadas. A grandeza residual em VA, anteriormente utilizada como critério, continua sendo gravada na variável `difs` para o gráfico de convergência apresentado na seção 5.2.7, mas perdeu a função de parada.

Concomitantemente, identificou-se uma condição de não-terminação no laço externo de compensação de barras de tensão controlada, herdada da versão de Ferreira (2021) e relevante para o caminho do EPRI. O laço enquanto da compensação PV atualizava a variável `erro_max_gd` somente quando o tipo do gerador era PV. Nos circuitos do EPRI, em que não há `EnergySchedulingType.PV` declarado, o vetor de geradores é vazio, e `erro_max_gd` permanecia no valor inicial superior à tolerância pelo laço inteiro, configurando laço infinito. A correção, aplicada em `Program.cs:L1201–L1204`, consiste em inicializar `erro_max_gd = 0` antes do laço quando o vetor de geradores é nulo ou vazio, condição que se reduz à única passagem pelo corpo do laço para os circuitos do EPRI e preserva o comportamento original para o caminho do sistema-teste de 135 barras.

4.8 TRADUÇÃO CIM $\rightarrow$ REDE

A costura entre as duas camadas do programa apoia-se em três classes-espelho introduzidas neste trabalho: `DadosZJRamoCimACLine`, `DadosSCimCarga` e `DadosTrafoCim`. Cada uma carrega o subconjunto de atributos eletricamente úteis derivados do CIM, na mesma tipagem das classes elétricas: `Complex[3][3]` para a matriz Z_{abc} do `ACLineSegment`, `Complex[3]` para o vetor S do `EnergyConsumer` agregado por `TopologicalNode`, e o conjunto de campos U_1 , U_2 , S_{base} , R , X , G , B e `relacaoTransformacao` para o `PowerTransformer`. Essas classes vivem na camada CIM (CPDSM) e são consumidas pela camada elétrica (Rede), funcionando como interface explícita entre as duas. A indexação adotada é estrita: dicionários `Dictionary<string, T>` com chave igual ao `mRID` da instância CIM correspondente (`ACLineSegment.mRID`, `EnergyConsumer.mRID`, `TopologicalNode.mRID` e `PowerTransformer.mRID`). A escolha tem três efeitos diretos: (i) torna o acesso $O(1)$ em ambas as direções; (ii) torna explícita, na linguagem de programação, a fronteira entre a representação CIM e a representação elétrica; e (iii) permite a auditoria dos quatro relatórios em texto descritos na seção 4.9 a partir dos `mRID` originais do XML CIM.

A inicialização da varredura merece um comentário específico relacionado ao

EnergySource. O parser CIM lê, para cada EnergySource, os atributos voltageMagnitude, voltageAngle, r , x , r_0 e x_0 (Program.cs:L1573–L1579), valores explicitamente declarados no ckt7_CIM100x.XML e no ckt24_CIM100x.XML. O motor de cálculo, na inicialização da raiz da árvore radial em converterCimParaRede, utiliza, contudo, apenas o atributo nominalVoltage da fonte (Program.cs:L440). Os demais atributos do EnergySource, embora carregados pelo parser, permanecem disponíveis no objeto dsm.energySource sem composição efetiva da árvore de cálculo. A consequência dessa simplificação é o pequeno resíduo de tensão de $\approx +1\%$ observado no primário na comparação com o OpenDSS (Capítulo 5), e a sua superação integra o primeiro item dos trabalhos futuros (seção 6.3).

O método converterCimParaRede da classe Rede (Program.cs:L426–L912) realiza a costura final entre as duas camadas do programa. Os passos executados podem ser resumidos como segue. (i) Instanciamento das barras: cada TopologicalNode lido do CIM gera uma Barra, cujos atributos Nome e mRID herdam o IdentifiedObject.name e o identificador do nó, respectivamente; a potência S é inicializada com os valores agregados em noSPorTopologicalNodeMrid, conforme (4.1). (ii) Instanciamento dos ramos: cada ACLineSegment, PowerTransformer, Switch, Disconnecter, Fuse OU Breaker gera um Ramo, cujos campos zRamo, Ym e relacaoTransformacao são preenchidos a partir dos dicionários da camada CIM; para chaves fechadas, atribui-se $Z = 0$; para chaves abertas, o ramo é descartado. (iii) Orientação *De/Para*: a orientação dos ramos é deduzida por DFS a partir do EnergySource, replicando o resultado de processTopology, com varredura BFS auxiliar em pontos de fronteira para resolver ambiguidades. (iv) Propagação de $V_{nominal}$: a tensão nominal de cada barra, em Volts, é propagada por BFS desde a raiz, com transição abrupta para o ratedU do enrolamento secundário ao atravessar um ramo de transformador. Esse atributo alimenta a inicialização das tensões da varredura.

4.9 RELATÓRIOS EM DISCO

Concluído o fluxo de potência, o programa grava quatro arquivos no diretório de execução, todos no formato texto delimitado para leitura programática externa. O método imprimirValor (Program.cs:L1228–L1337) escreve resultadoPF_<nomeCircuito>.txt, com tensões fase-neutro $|V_a|$, $|V_b|$, $|V_c|$ em Volts e respectivos ângulos, além das tensões fase-fase $|V_{ab}|$, $|V_{bc}|$, $|V_{ca}|$ derivadas. Cada linha apresenta o mRID e o Nome da

barra, este último coincidente com o identificador exportado pelo OpenDSS, permitindo o cruzamento *bus-a-bus* discutido no Capítulo 5.

Os três relatórios complementares (`MatrizesZ_<nomeCircuito>.txt`, `Potencias-Carga_<nomeCircuito>.txt` e `Transformadores_<nomeCircuito>.txt`), gerados pelos métodos estáticos `imprimirMatrizesZ`, `imprimirPotenciasCarga` e `imprimirTransformadores` da classe `CPDSM`, contêm respectivamente as matrizes Z_{abc} por `ACLineSegment.mRID`, as potências S_a , S_b , S_c por `EnergyConsumer.mRID` e por `TopologicalNode.mRID`, e os parâmetros físicos dos transformadores (U_1 , U_2 , S_{base} , R , X , G , B , conexões dos enrolamentos, caminho A ou B). Esses arquivos cumprem dois propósitos complementares: (i) constituem registro auditável dos dados eletricamente úteis derivados do XML CIM, e (ii) permitem a substituição parcial do motor de fluxo por um simulador externo qualquer, sem necessidade de reescrita do parser.

A escolha das colunas-chave `mRID` e `Nome` na primeira linha de cada arquivo merece justificativa explícita, dado seu papel central na comparação numérica do Capítulo 5. O `mRID` é o identificador universal único da instância CIM, presente no atributo `rdf:ID` do elemento XML, e funciona como chave primária estável para o cruzamento entre os quatro relatórios produzidos pelo programa. O `Nome`, herdado do atributo `cim:IdentifiedObject.name` do `TopologicalNode`, é o rótulo legível da barra (por exemplo `SOURCEBUS`, `318405` ou `S1X_1001577` nos circuitos do EPRI) e coincide com o identificador exportado pelo OpenDSS em seu relatório de tensões nodais. Essa coincidência não é fortuita, dado que o tradutor do OpenDSS para CIM, mantido pelo EPRI, adota o `name` do `TopologicalNode` como nome de barra. A presença explícita de ambas as chaves em `resultadoPF_*` viabiliza o cruzamento *bus-a-bus* por simples operação `join` em planilha ou em `pandas`, sem necessidade de reabertura do XML para resolução de referências.

5 RESULTADOS E COMPARAÇÃO COM OPENDSS

Apresenta-se neste capítulo a avaliação experimental do algoritmo descrito no Capítulo 4. A estratégia adotada conjuga dois eixos de verificação: a regressão de comportamento sobre o sistema-teste de 135 barras do LAPSEE, herdado de Ferreira (2021), e a comparação numérica detalhada com o simulador OpenDSS nos três circuitos de teste do EPRI (ckt5, ckt7 e ckt24), com análise estatística aprofundada do ckt7 e síntese consolidada dos demais na seção 5.3.

5.1 VALIDAÇÃO NO SISTEMA-TESTE DE 135 BARRAS

A primeira etapa de verificação consistiu em reexecutar o caminho de entrada do XML proprietário do LAPSEE (Opção 1 do menu principal), de modo a confirmar que as alterações introduzidas neste trabalho (em particular o novo critério de convergência em tensão e as guardas anti-*NaN* descritas na seção 4.7) preservam o comportamento operacional do motor herdado de Ferreira (2021). O sistema-teste é composto por 135 barras de carga acopladas a uma subestação de 13,8 kV (linha-linha), totalizando 136 barras topológicas, 135 ramos exclusivamente compostos por trechos *ACLineSegment* e nenhum transformador interno ao alimentador.

A execução convergiu em oito iterações, com resíduo final $\max|\Delta S| = 3,77 \times 10^{-4}$ VA, partindo de $3,45 \times 10^4$ VA na iteração inicial. A taxa de redução próxima de uma ordem de grandeza por iteração reproduz o padrão de decaimento exponencial característico da varredura *backward/forward* em redes radiais bem condicionadas, conforme demonstrado em (3.4) e (3.5) de De Souza, Leite e Reiz (2024).

A Tabela 2 apresenta as tensões fase-neutro em pu de um subconjunto representativo de barras, normalizadas pela tensão de referência da subestação $V_{SE} = 7967,43$ V (correspondente a $13,8/\sqrt{3}$ kV).

Barra	$\ V_a\ $ (pu)	$\ V_b\ $ (pu)	$\ V_c\ $ (pu)	$\angle V_a$ (°)
1 (subestação)	1,0000	1,0000	1,0000	0,000
20	0,9556	0,9559	0,9635	-1,056
60	0,9469	0,9473	0,9563	-1,251
100	0,9416	0,9420	0,9519	-1,375
135 (extremidade)	0,9403	0,9407	0,9508	-1,400

Tabela 2 – Tensões em pu obtidas pelo programa no sistema-teste de 135 barras (base $V_{SE} = 7967,43$ V LN).

Fonte: Próprio autor.

Três aspectos da Tabela 2 sustentam a regressão neutra. O primeiro é a monotonia da queda de tensão da subestação até a extremidade do alimentador, com valores que decrescem de 1,000 pu na barra 1 até 0,940 pu na barra 135 (queda total de aproximadamente 6% nas fases A e B e de 4,9% na fase C), valores compatíveis com a operação de alimentadores de média tensão urbanos sem regulação ativa. O segundo é o equilíbrio entre as fases A e B, cuja diferença máxima observada não ultrapassa 0,04 pu, ao passo que a fase C apresenta tensão sistematicamente mais elevada, comportamento herdado do desbalanço de cargas registrado no XML original do LAPSEE. O terceiro é o avanço progressivo do ângulo da fase A, que parte da referência 0° na subestação e atinge $-1,400^\circ$ na extremidade, defasagem coerente com a queda angular esperada para um alimentador radial com carregamento da ordem de 7 MVA.

A próxima seção concentra-se na avaliação central do trabalho, a comparação com o OpenDSS no circuito ckt7, em que a base de comparação foi montada de forma sistemática e os números são reproduzíveis a partir do CSV anexado em `planilhas finais/Comparacao_ckt7_BarraABarra.csv`.

5.2 CIRCUITO CKT7 DO EPRI: ANÁLISE DETALHADA

5.2.1 Características do circuito

O circuito ckt7, distribuído pelo EPRI no perfil CIM 100 (ELECTRIC POWER RESEARCH INSTITUTE, [s.d.]c), representa um alimentador de média tensão real de 12,47 kV, com cerca de 1255 nós topológicos, 5694 unidades consumidoras e potência instalada da ordem de 19,3 MVA. A Tabela 3 sintetiza as principais características elétricas.

Parâmetro	Valor
Tensão nominal do primário	12,47 kV (linha-linha)
Número de nós topológicos	1255
Número de consumidores (EnergyConsumer)	5694
Potência instalada agregada	19.320 kVA
Caminho CIM dos transformadores	Caminho B (TransformerTank + ensaios)

Tabela 3 – Características principais do circuito ckt7 do EPRI.

Fonte: Próprio autor.

O alimentador apresenta seções em três níveis de tensão distintos: primário em 12,47 kV linha-linha, secundários trifásicos em 208 V (sistema Y) e secundários split-phase em 240 V (sistema monofásico de duas fases). A presença de barras em todos esses níveis impõe a normalização adaptativa por classe de tensão discutida na seção 4.8 e implementada também no *script* de análise.

5.2.2 Procedimento de execução

A reprodução dos resultados envolve quatro etapas. Inicialmente, executa-se o aplicativo ConsoleApp.exe e seleciona-se a Opção 2 do menu, seguida de ckt7. Em segundo lugar, os arquivos resultadoPF_ckt7_CIM100x.txt, MatricesZ_ckt7_CIM100x.txt, PotenciasCarga_ckt7_CIM100x.txt e Transformadores_ckt7_CIM100x.txt são gerados em ConsoleAppAli/bin/Debug/. Em terceiro lugar, a referência do OpenDSS é obtida pelos comandos Export Voltages e Export Currents (DUGAN; MONTENEGRO, 2020), que produzem, respectivamente, as magnitudes fase-neutro por barra e as correntes de envio por elemento, com o circuito resolvido na mesma carga nominal ingerida pelo programa. O pareamento *bus-a-bus* usa como chave o atributo cim:Topological-Node.IdentifiedObjectName, coincidente com o identificador de barra do simulador; a comparação é insensível a maiúsculas, pois o OpenDSS exporta os nomes em caixa alta enquanto o CIM os grava em minúsculas, e a fase é identificada pelo nó (Node) de cada coluna do export (1 para A, 2 para B, 3 para C). Por fim, os *scripts* scripts/analisa_ckt7.py e scripts/analisa_corrente.py processam essas saídas, calculam as estatísticas de tensão e de corrente e geram as cinco figuras apresentadas a seguir.

5.2.3 Estatísticas globais

A Tabela 4 reúne as estatísticas das diferenças percentuais $V_{\text{programa}} - V_{\text{DSS}}$ no primário de 12,47 kV, por fase e no conjunto das três fases, com a fase identificada pelo nó do Export Voltages. Os valores foram gerados pelo *script* analisa_ckt7.py e exportados em tabela_stats_ckt7.csv. O sinal positivo indica tensão do programa ligeiramente acima da reportada pelo simulador.

Fase	Nº de leituras	Erro médio (%)	Erro mediano (%)	Mínimo (%)	Máximo (%)
A	241	+1,27	+1,01	+0,50	+2,86
B	204	+1,26	+0,91	+0,42	+2,86
C	203	+1,17	+0,80	+0,30	+2,84

Fase	Nº de leituras	Erro médio (%)	Erro mediano (%)	Mínimo (%)	Máximo (%)
A+B+C	648	+1,24	+0,93	+0,30	+2,86

Tabela 4 – Diferenças percentuais $V_{\text{programa}} - V_{\text{DSS}}$ no primário 12,47 kV do ckt7, por fase e no conjunto. Valores positivos indicam tensão do programa acima da do simulador.

Fonte: Próprio autor.

Três observações se desprendem da Tabela 4. Em primeiro lugar, a diferença mediana no conjunto das três fases é de +0,93%, com média de +1,24% e todas as 648 leituras contidas na faixa estreita de +0,30% a +2,86%, o que situa o programa próximo do simulador e ligeiramente acima dele em todo o primário. Em segundo lugar, as três fases concordam entre si, com medianas entre +0,80% e +1,01%, de modo que o pequeno deslocamento positivo é uniforme e não específico de uma fase. Em terceiro lugar, a fase A reúne mais leituras (241) que B e C (204 e 203), por ser a única presente também nos ramais monofásicos derivados do primário, sem que isso altere a ordem de grandeza do resíduo.

A análise estratificada pelos quatro níveis de tensão do circuito, apresentada na Tabela 5, mostra como esse resíduo se distribui da fonte até os secundários, com a fase identificada pelo nó do export e o secundário split-phase 240 V pareado por magnitude.

Nível (V_{LL})	Nº de leituras	Erro médio (%)	Erro mediano (%)	Faixa (%)
115 kV (fonte)	3	+3,16	+3,16	[+3,15, +3,18]
12,47 kV (primário)	648	+1,24	+0,93	[+0,30, +2,86]
240 V (secundário split-phase)	662	+1,45	+1,45	[-0,47, +3,64]
208 V (secundário Y)	1089	+1,69	+1,79	[-0,28, +7,28]

Tabela 5 – Diferenças percentuais $V_{\text{programa}} - V_{\text{DSS}}$ no ckt7, estratificadas pelos quatro níveis de tensão.

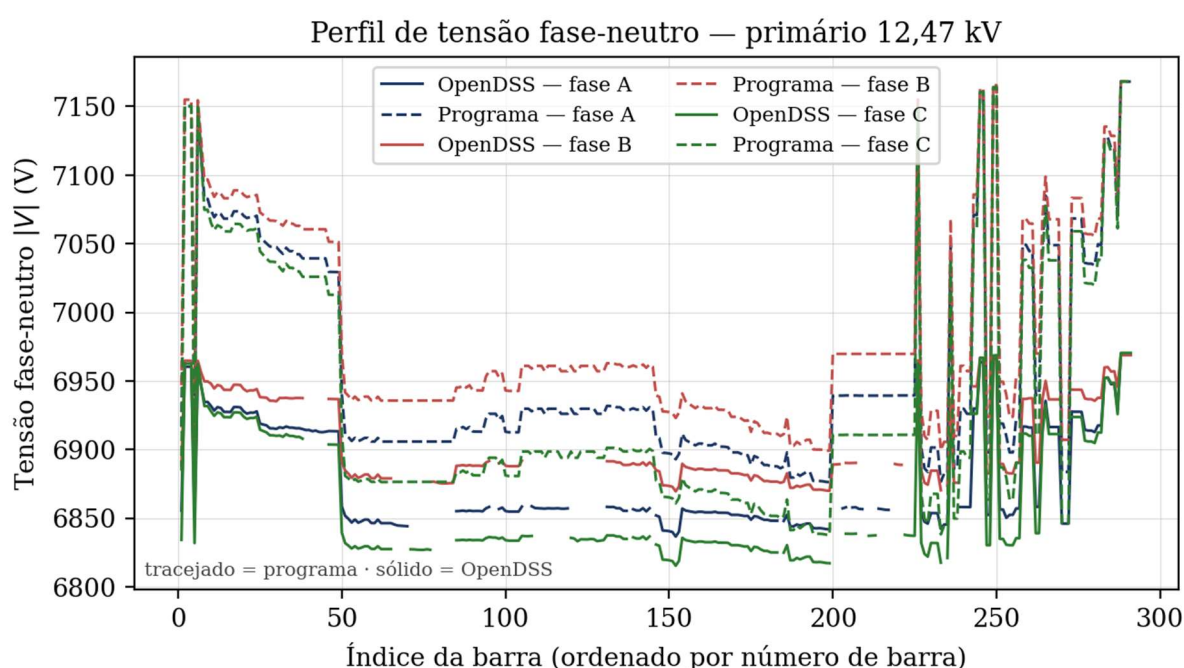
Fonte: Próprio autor.

O resíduo é maior na barra de 115 kV, a fonte da subestação (+3,16% de mediana), e atenua-se para +0,93% no primário de 12,47 kV. Esse comportamento decorre da modelagem da fonte, discutida em 5.2.8: o programa fixa o equivalente da subestação na tensão nominal de 115 kV (1,0 pu), enquanto o V_{source} do OpenDSS,

declarado em 1,05 pu atrás de uma impedância de Thévenin, tem o terminal rebaixado para cerca de 0,969 pu sob a carga nominal, o que coloca o programa acima do simulador na fonte; ao longo do alimentador, a diferença atenua-se para a ordem de +1% no primário, conforme detalhado na seção 5.2.8. Os secundários permanecem na mesma faixa, com medianas de +1,45% em 240 V e +1,79% em 208 V, o que confirma que o resíduo se mantém pequeno e positivo até as pontas do circuito, sem amplificação pelos transformadores de distribuição.

5.2.4 Perfil de tensão do primário

A Figura 5 apresenta o perfil de tensão fase-neutro em Volts ao longo das barras do primário de 12,47 kV, dispostas pelo número de barra. As três fases aparecem distinguidas por cor; a curva sólida representa o OpenDSS e a tracejada, o programa proposto.



Perfil de tensão fase-neutro em Volts no primário 12,47 kV do ckt7, por fase: programa proposto (tracejado) e OpenDSS (sólido).

Figura 5 – Perfil de tensão fase-neutro no primário 12,47 kV do ckt7.

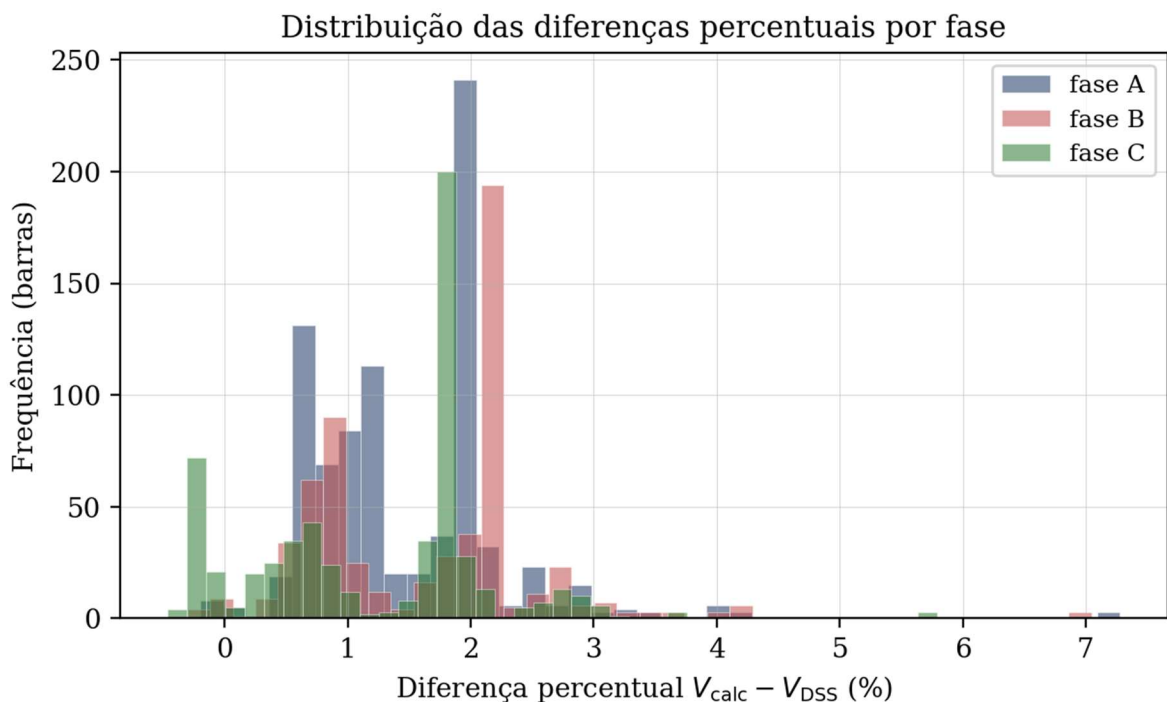
Fonte: Próprio autor.

A leitura da Figura 5 mostra as duas curvas praticamente sobrepostas ao longo do tronco do alimentador, com o programa situado de forma consistente um pouco acima do OpenDSS. O afastamento típico é da ordem de 60 a 90 V sobre uma tensão fase-neutro próxima de 7000 V, compatível com a mediana de +0,93% da Tabela 4.

Esse pequeno deslocamento positivo reflete a modelagem da fonte (seção 5.2.8): o programa mantém o equivalente da subestação na tensão nominal, ao passo que o terminal do V_{source} do OpenDSS, com força eletromotriz de 1,05 pu atrás de impedância, opera abaixo de 1,0 pu sob a carga nominal. As oscilações na porção direita do perfil correspondem aos ramais e às barras de extremidade, nos quais a presença parcial das fases e a maior dispersão das cargas tornam o traçado menos suave, sem alterar o padrão de proximidade entre os dois cálculos.

5.2.5 Distribuição das diferenças percentuais

A Figura 6 apresenta o histograma das diferenças percentuais $V_{programa} - V_{DSS}$ por fase, abrangendo as barras pareadas em todos os níveis de tensão do circuito.



Histograma das diferenças percentuais $V_{programa} - V_{DSS}$ por fase no ckt7, com sobreposição das três distribuições.

Figura 6 – Distribuição das diferenças percentuais por fase no ckt7.

Fonte: Próprio autor.

Dois aspectos se destacam na Figura 6. Primeiro, as três distribuições são positivas e concentradas, com a maior parte das leituras entre +0,5% e +3%, em acordo com as medianas da Tabela 5. Segundo, a leve estrutura bimodal acompanha os níveis de tensão: o modo em torno de +0,9% reúne as barras do primário, ao passo que o modo próximo de +1,8% corresponde aos secundários 208 V, ligeiramente mais

afastados por incorporarem a relação fixa dos transformadores de distribuição. A concentração das diferenças num intervalo estreito e positivo confirma que o programa acompanha o simulador ao longo de todo o alimentador, com deslocamento pequeno e homogêneo.

5.2.6 Dispersão entre os dois simuladores

A Figura 7 apresenta o diagrama de dispersão entre as tensões calculadas pelos dois simuladores, com a reta $y = x$ servindo de referência geométrica.

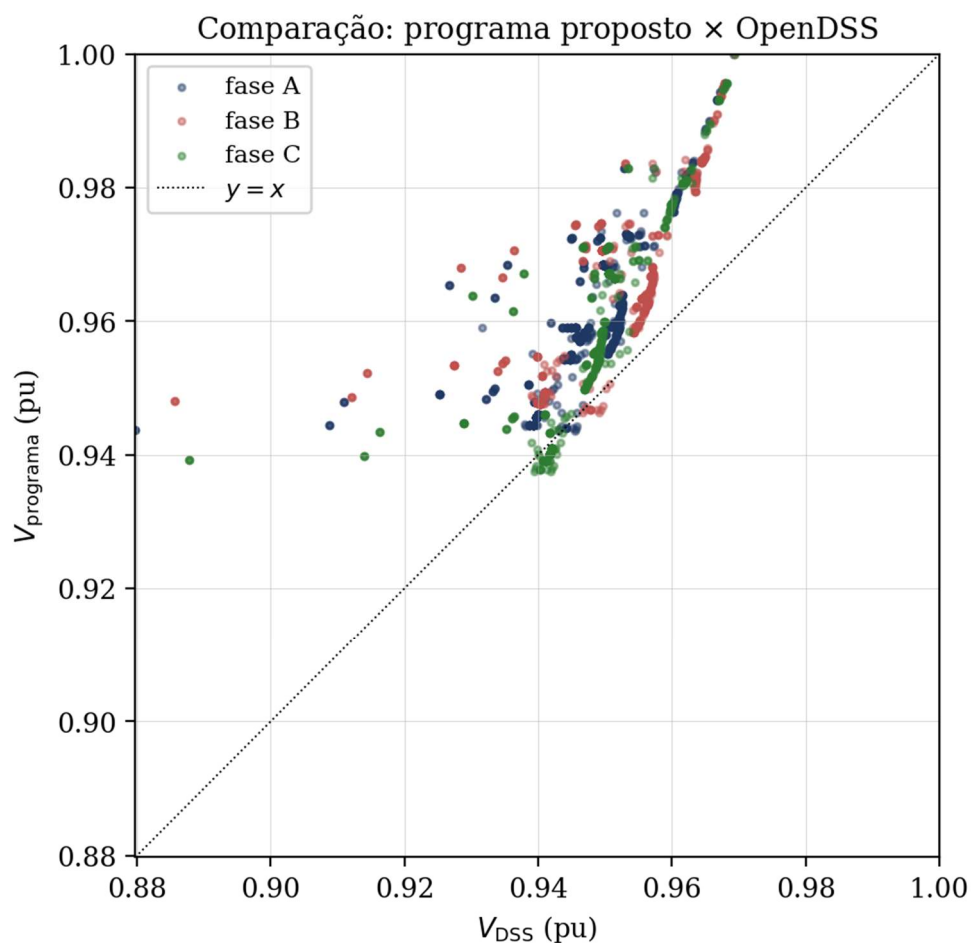


Diagrama de dispersão das tensões em pu calculadas pelo programa proposto (eixo y) contra as do OpenDSS (eixo x). A reta $y = x$ é a referência de igualdade.

Figura 7 – Dispersão $V_{\text{programa}} \times V_{DSS}$.

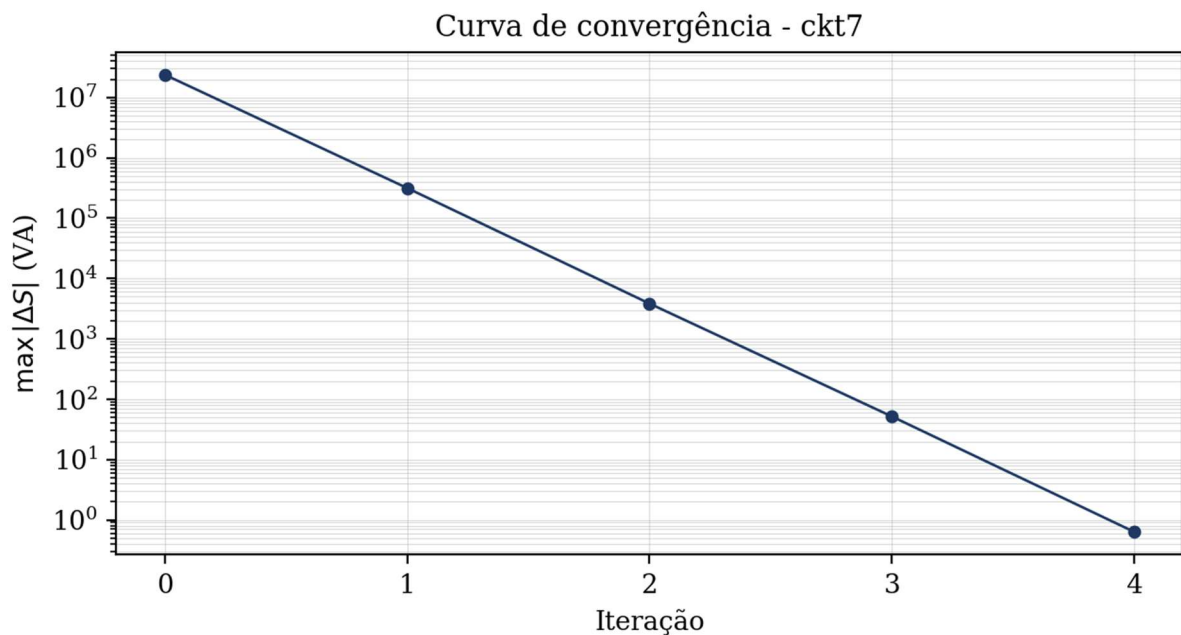
Fonte: Próprio autor.

A nuvem de pontos da Figura 7 acompanha de perto a reta $y = x$ e situa-se ligeiramente acima dela, em coerência com o deslocamento positivo descrito nas seções anteriores: para um mesmo valor de V_{DSS} , a tensão do programa é um pouco maior. Os pontos concentram-se na faixa de 0,94 a 0,97 pu, onde se acumulam as

barras do tronco do primário e dos secundários sob carga nominal, e a dispersão em torno da reta é pequena, o que indica boa concordância ponto a ponto entre os dois cálculos.

5.2.7 Curva de convergência

A Figura 8 apresenta o comportamento do resíduo de potência $\max|\Delta S|$ ao longo das iterações da varredura *backward/forward*, conforme registrado no bloco “DIFERENÇA DE POTÊNCIA POR ITERAÇÃO” do relatório `resultadoPF_ckt7_CIM100x.txt`. O resíduo de potência é mantido no relatório como métrica auxiliar de diagnóstico, ainda que o critério de parada efetivamente empregado pelo método `calculatePF` seja $\max|\Delta V| < \text{tol}_V$, conforme apresentado na seção 4.7 e justificado pelos comentários técnicos do código.



Curva de convergência: $\max|\Delta S|$ por iteração no ckt7, em escala logarítmica.

Figura 8 – Curva de convergência do ckt7.

Fonte: Próprio autor.

A trajetória observada confirma o comportamento esperado para a varredura *backward/forward* aplicada a um alimentador radial bem condicionado: o resíduo de potência decai de $2,38 \times 10^7$ VA na iteração inicial para $6,34 \times 10^{-1}$ VA na quinta iteração, com taxa de redução próxima de uma ordem de grandeza por iteração e decaimento aproximadamente linear em escala logarítmica. A convergência em cinco iterações é coerente com o desempenho relatado por De Souza, Leite e Reiz (2024) para

o mesmo algoritmo aplicado a redes de distribuição reais de grande porte, e demonstra que a substituição do critério em potência pelo critério em tensão (seção 4.7) não introduz penalidade no número de iterações necessárias.

5.2.8 Discussão dos desvios

As seções 5.2.3 a 5.2.7 mostram concordância estreita entre os simuladores e isolam duas diferenças sistemáticas de origem conhecida, uma em magnitude e outra em ângulo, além de fatores de menor monta.

(a) Magnitude: modelagem da fonte da subestação. O pequeno deslocamento positivo da tensão, com mediana de +0,93% no primário e +3,16% na barra de 115 kV (Tabela 5), origina-se da forma como cada ferramenta representa o equivalente da subestação. O algoritmo proposto inicializa a raiz da varredura com o `nominalVoltage` de 115 kV, ou seja, 1,0 pu (Program.cs linha 440); o parser lê também os atributos `voltageMagnitude`, `voltageAngle`, `r`, `x`, `r0` e `x0` do `EnergySource` (Program.cs linhas 1573 a 1579), que permanecem descartados pelo motor de cálculo no estágio atual. O OpenDSS, por sua vez, modela a fonte como um `Vsource` com força eletromotriz declarada em `cim:EnergySource.voltageMagnitude` de 120.750 V (razão exata de 1,05 sobre o `nominalVoltage`) atrás de uma impedância de Thévenin ($r = 2,314 \Omega$, $x = 28,52 \Omega$ no `ckt7_CIM100x.XML`). Sob a carga nominal, o terminal desse `Vsource` é rebaixado para cerca de 0,969 pu, valor inferior a 1,0 pu, o que mantém o programa um pouco acima do simulador na fonte; ao longo do primário, esse resíduo atenua-se para a ordem de +1%. Esse valor na fonte não é fortuito: o resíduo em qualquer barra é, por definição, $V_{\text{programa}}/V_{\text{DSS}} - 1$, e na barra de 115 kV o programa mantém V_{programa} na tensão nominal (1,0 pu, ou 66.395 V fase-neutro), enquanto o OpenDSS reporta o terminal rebaixado para 0,969 pu sob a carga nominal; a razão entre as duas tensões reproduz a mediana de +3,16% da Tabela 5. O afundamento do terminal para 0,969 pu corresponde à força eletromotriz declarada de 1,05 pu reduzida pela queda na impedância de Thévenin sob carga. Modelar o `EnergySource` como ramo Thévenin equivalente, aproveitando os atributos já lidos pelo parser, integra a lista de trabalhos futuros (seção 6.3). Registre-se que o `ckt7` não possui reguladores de tensão com `RatioTapChanger`: os `RegulatingControl` do modelo atuam sobre dois `LinearShuntCompensator`, de modo que a regulação por *tap* não participa do quadro do `ckt7`. Esses dois capacitores, energizados pelo OpenDSS e mantidos desligados pelo programa,

elevam a tensão do simulador a jusante e atenuam o resíduo de +3,16% da fonte para +0,93% no primário.

(b) Ângulos: deslocamento por *vector group*. A análise dos atributos `AngA_DSS`, `AngB_DSS` e `AngC_DSS` do export revela uma defasagem média de $\approx -35^\circ$ entre os ângulos da fase A reportados pelo OpenDSS e os calculados pelo programa. O valor é compatível com o deslocamento angular de 30° característico das conexões Δ -Y dos transformadores de distribuição, somado às pequenas variações angulares acumuladas ao longo dos ramais. A inspeção direta do `ckt7_CIM100x.XML` confirma o *vector group* Δ -Y como mecanismo dominante: identificam-se 33 ocorrências do atributo `cim:PowerTransformerEnd.phaseAngleClock = 1` (correspondente a um deslocamento de 30° entre primário e secundário) e 33 ocorrências de `phaseAngleClock = 0`, configuração compatível com bancos trifásicos compostos por um enrolamento Δ e um enrolamento Y. Como o método `calculatePF` trata o transformador apenas pelo módulo da relação a , sem aplicar o deslocamento de fase definido pelo `connectionKind`, o ângulo de saída fica preso à referência da subestação, enquanto o OpenDSS desloca cada barra conforme o `vectorGroup` acumulado no caminho até a fonte. Os ângulos das fases B e C apresentam dispersão maior por refletirem também eventuais permutações de fase em ramais derivados, mas o padrão é coerente com a explicação geral.

(c) Outras simplificações. Três fatores adicionais contribuem em menor grau para os desvios observados: (i) o efeito *shunt* das linhas e o ramo magnetizante dos transformadores estão desativados na varredura, conforme discutido em (4.4) e (4.8); (ii) cargas registradas no CIM apenas como totais trifásicos são distribuídas uniformemente entre as fases por (4.1), gerando desbalanço residual nos ramais monofásicos; (iii) eventuais inconsistências entre a topologia exportada pelo OpenDSS e a topologia processada por `processTopology` foram inspecionadas amostralmente, sem evidência de erro estrutural, visto que os nós do export puderam ser pareados com o `IdentifiedObject.name` do CIM.

Tomados em conjunto, os fatores (a), (b) e (c) explicam de forma consistente a estrutura das diferenças e permitem balizar as direções de pesquisa futuras enumeradas no Capítulo 6. O resíduo de magnitude é determinístico e tratável pela modelagem do `EnergySource` como ramo Thévenin; o desvio angular é igualmente determinístico e tratável pela incorporação do `vectorGroup`; os demais fatores têm magnitude residual.

5.2.9 Corrente de ramo

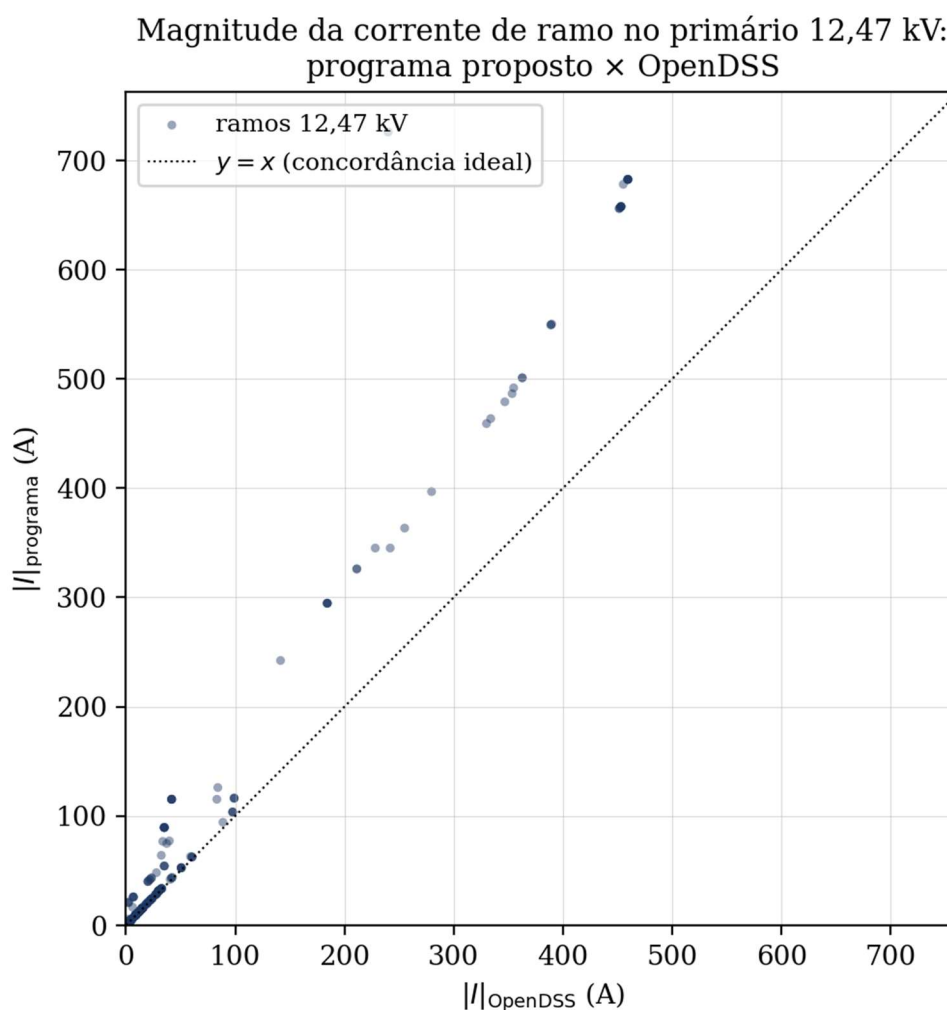
A verificação da corrente complementa a da tensão. A métrica adotada é a magnitude de ramo $|I| = \sqrt{I_a^2 + I_b^2 + I_c^2}$, independente de fase, comparada elemento a elemento entre o programa e o Export Currents do OpenDSS, com piso de $|I|_{\text{DSS}} \geq 1$ A para descartar ramos sem fluxo apreciável e com linhas separadas de transformadores. O ckt7 é uma rede radial: o valor bruto $E - N + 1 = 64$ conta apenas arestas paralelas (32 pares de barras ligados por mais de um ramo direto e os três transformadores de subestação sobre o mesmo par de barras), e, após fundir esses paralelos, a rede não apresenta laços. A Tabela 6 reúne as estatísticas por conjunto, geradas pelo *script analisa_corrente.py*.

Conjunto	Nº de ramos	Erro mediano (%)	Erro médio (%)
Linhas (todas)	1116	5,37	15,93
Linhas no primário 12,47 kV	250	3,79	36,68
Transformadores	129	-26,31	55,11
Todos os ramos	1245	5,35	19,99

Tabela 6 – Diferenças percentuais da magnitude de corrente de ramo $|I|$ no ckt7, separadas em linhas e transformadores.

Fonte: Próprio autor.

A Figura 9 apresenta a dispersão da magnitude de corrente nos ramos do primário 12,47 kV, com a reta $y = x$ como referência de igualdade.



Magnitude da corrente de ramo no primário 12,47 kV do ckt7: programa proposto (eixo y) contra OpenDSS (eixo x), em escala linear, com a reta $y = x$.

Figura 9 – Magnitude da corrente de ramo no primário 12,47 kV do ckt7.

Fonte: Próprio autor.

A leitura conjunta da tabela e da figura indica boa concordância no corpo da distribuição: a mediana é de 5,4% no conjunto das linhas e de 3,8% nos ramos do primário, e a mediana das magnitudes (14,7 A no programa contra 13,3 A no OpenDSS) praticamente coincide. A média é maior, de 15,9% nas linhas, por efeito de uma cauda concentrada nas poucas barras junto à subestação: ali os 32 pares de ramos paralelos e os três transformadores de subestação em paralelo são percorridos, na varredura radial, por um único caminho equivalente, que acumula a corrente dos paralelos e chega a destoar por um fator de até cerca de três (o percentil 95 da magnitude sobe de 184 A no OpenDSS para 295 A no programa). Esse efeito restringe-se aos primeiros níveis da árvore e não se propaga para o restante do alimentador, em que os pontos se alinham à reta $y = x$. Os transformadores de distribuição aparecem

à parte, com mediana de -26% , atribuível à diferença de modelagem do ramo do transformador entre as duas ferramentas, e não ao fluxo de corrente no primário.

5.3 CIRCUITOS CKT5 E CKT24: SÍNTESE

Os circuitos ckt5 e ckt24, também distribuídos no perfil CIM 100 do EPRI, foram processados pelo mesmo pipeline descrito na seção 5.2.2 e tiveram suas saídas de tensão e de corrente pareadas elemento a elemento com o OpenDSS. A Tabela 7 consolida as estatísticas globais das diferenças percentuais de tensão $V_{\text{programa}} - V_{\text{DSS}}$ para os dois circuitos, com a fase A computada sobre o conjunto completo de barras e as fases B e C restritas, conforme o caso, aos trechos trifásicos e aos secundários 208 V.

Circuito	Fase	Nº de barras	Erro médio (%)	Erro máx. $ \cdot $ (%)	Desvio-padrão (%)
ckt5	A	2996	-0,02	5,01	1,30
ckt5	B	227	-1,08	2,87	0,74
ckt5	C	208	-1,35	2,64	0,69
ckt24	A	6058	-5,40	9,18	0,93
ckt24	B	748	-6,60	9,83	0,34
ckt24	C	716	-6,37	9,59	0,30

Tabela 7 – Estatísticas globais das diferenças percentuais de tensão $V_{\text{programa}} - V_{\text{DSS}}$ para os circuitos ckt5 e ckt24.

Fonte: Próprio autor.

Os três circuitos ordenam-se pela tensão que o OpenDSS mantém na cabeça do alimentador, grandeza que governa o sinal e a ordem de grandeza do resíduo de tensão, dado que o programa fixa a raiz da varredura em 1,0 pu. Medida na cabeça do primário, essa tensão vale 1,000 pu no ckt5, 0,968 pu no ckt7 e 1,039 pu no ckt24, e reproduz o sinal de cada resíduo. O ckt5, de 12,47 kV, declara a fonte em 1,05 pu com impedância desprezível e não possui regulador de tensão (nenhum `RatioTapChanger`); o transformador da subestação entrega a cabeça do primário na tensão nominal, que coincide com a referência do programa, e o erro médio na fase A é de apenas $-0,02\%$, com máximo de $5,01\%$ em barras secundárias específicas. O ckt24, de 34,5 kV, tem fonte rígida, cuja impedância de Thévenin reduzida quase não rebaixa o terminal, e a subestação dispõe de um regulador de tensão sob carga, que eleva o barramento regulado para cerca de 1,04 pu, além de três bancos de capacitores; o programa, que mantém a fonte em 1,0 pu e não modela esses controles (seção 6.2),

situa-se entre $-5,40\%$ e $-6,60\%$ abaixo do OpenDSS. A maior parte dessa diferença, da ordem de $-3,7\%$, decorre da própria tensão de cabeça (1,0 pu do programa contra 1,039 pu do OpenDSS); os capacitores e a compensação de queda de linha acrescentam o restante, crescente em direção aos secundários. O ckt7, sem regulador e com a fonte rebaixada sob carga, coloca o programa acima do simulador, com o pequeno resíduo de $+0,93\%$ (seção 5.2.3). A baixa dispersão em todos os casos, com desvio-padrão inferior a 1,3%, confirma o caráter sistemático das diferenças e afasta a hipótese de instabilidade numérica do método.

A síntese de corrente, na Tabela 8, fornece a verificação que independe da regulação de tensão, pois compara o fluxo de corrente de ramo pela magnitude $|I|$, separando linhas de transformadores.

Circuito	Nº de linhas	Erro mediano, linhas (%)	Nº de trafos	Erro mediano, trafos (%)
ckt5	2286	3,80	495	-27,09
ckt7	1116	5,37	129	-26,31
ckt24	5023	5,05	590	-25,99

Tabela 8 – Diferenças percentuais medianas da magnitude de corrente de ramo $|I|$ nos três circuitos, separadas em linhas e transformadores.

Fonte: Próprio autor.

A corrente de ramo das linhas concorda na mesma faixa nos três circuitos, com mediana de 3,8% no ckt5, 5,4% no ckt7 e 5,0% no ckt24, o que demonstra que a varredura reproduz o fluxo de corrente dos alimentadores independentemente da regulação de tensão ativa. Os transformadores aparecem à parte, com mediana próxima de -26% nos três circuitos, padrão homogêneo que aponta para a diferença de modelagem do ramo do transformador entre as duas ferramentas, e não para um erro de varredura.

A Tabela 9 reúne os parâmetros declarados da fonte de cada circuito, lidos do Master_ckt5.dss original (ELECTRIC POWER RESEARCH INSTITUTE, [s.d.]a) no caso do ckt5 e do atributo cim:EnergySource nos arquivos ckt7_CIM100x.XML e ckt24_CIM100x.XML no caso dos demais.

Circuito	pu declarado	V_{nominal} (kV)	$V_{\text{magnitude}}$ (kV)	r (Ω)	x (Ω)	r_0 (Ω)	x_0 (Ω)
ckt5	1,05	—	—	0	0,001	0	0,001
ckt7	1,05	115	120,75	2,314	28,52	0,0001	0,0001
ckt24	1,05	230	241,50	0,63	6,72	4,07	15,55

Tabela 9 – Parâmetros do EnergySource declarados para cada circuito do EPRI.

O ckt5 declara a fonte no formato `Master.dss` original; o ckt7 e o ckt24 declaram a fonte no `cim:EnergySource` do XML CIM 100. Magnitude $V_{\text{magnitude}}$ em valor absoluto (linha-linha).

Fonte: Próprio autor.

Os três circuitos declaram a fonte em 1,05 pu sobre a tensão nominal, com impedâncias de Thévenin que variam de desprezível no ckt5 ($x_1 = 10^{-3} \Omega$) a apreciável no ckt7 e no ckt24 ($x = 28,52 \Omega$ e $x = 6,72 \Omega$). Esses parâmetros, lidos pelo parser mas ainda não usados pelo motor de cálculo (seção 5.2.8), são reunidos aqui como referência para a modelagem futura do `EnergySource` como ramo Thévenin (seção 6.3). As diferenças entre os caminhos de modelagem do transformador (caminho A do `TransformerMeshImpedance` no ckt5 contra caminho B do `ShortCircuitTest` no ckt24) não se traduzem em descontinuidade estatística mensurável, o que valida o tratamento unificado dos dois caminhos no algoritmo da seção 4.6.

6 CONCLUSÃO

Apresentou-se, ao longo deste trabalho, a implementação e a validação experimental de um algoritmo de fluxo de potência trifásico para sistemas de distribuição modelados em CIM/DCIM, com base no perfil CIM 100 adotado pelo EPRI. O ponto de partida foi o motor de varredura *backward/forward* desenvolvido por Ferreira (2021) e refinado por Souza (2023), sobre o sistema-teste de 135 barras do LAPSEE. A esse ponto de partida foram acoplados quatro blocos próprios: a derivação das grandezas eletricamente úteis a partir do CIM bruto (matriz Z_{abc} das linhas, vetor S por barra, parâmetros físicos dos transformadores), a tradução do modelo CIM para a hierarquia Rede/Barra/Ramo, a adaptação da busca em profundidade para incorporar a relação de transformação a como elemento ativo da varredura, e a geração de relatórios formatados para o cruzamento *bus-a-bus* com o OpenDSS.

Adicionalmente, três correções foram aplicadas ao código herdado para viabilizar a execução nos circuitos do EPRI: a substituição do critério de convergência $|\Delta S| < \varepsilon$ pelo critério $\max|\Delta V| < \text{tol}_V$, indispensável para a operação em grandezas físicas SI; o confinamento da `CultureInfo.InvariantCulture` ao escopo do *parser* CIM, evitando a interpretação errônea do separador decimal sob *locale pt-BR*; e a conversão das propriedades `TransformerMeshImpedance` e `TransformerCoreAdmittance` de instância única para arrays, dado que os circuitos `ckt7` e `ckt24` contêm dezenas de cada classe.

A validação experimental no circuito `ckt7` do EPRI, conduzida em tensão e em corrente contra a saída do OpenDSS, registrou no primário de 12,47 kV diferença mediana de tensão de +0,93%, com o programa ligeiramente acima do simulador por manter o equivalente da subestação na tensão nominal enquanto o V_{source} do OpenDSS opera abaixo de 1,0 pu sob a carga nominal. A magnitude da corrente de ramo concordou com mediana de $\approx 3,8\%$ no primário e de $\approx 5,4\%$ no conjunto das linhas. A análise dos circuitos `ckt5` e `ckt24` reproduziu o mesmo quadro, com corrente de linha mediana de 3,8% e 5,0% respectivamente e diferenças de tensão compatíveis com a tensão que a fonte de cada circuito entrega na cabeça do alimentador (a rigidez da fonte e, no `ckt24`, o regulador da subestação). Os resultados confirmam a validade do algoritmo de varredura operando em grandezas físicas SI.

O modelo implementado neste trabalho apresenta cinco limitações reconhecidas, cuja superação configura direção natural de continuidade.

(a) Ausência de regulação ativa. A classe `RegulatingControl` do CIM, responsável pela parametrização do estado dos bancos de capacitores chaveados e dos *taps* dos reguladores de tensão, não é interpretada pelo programa. Todos os reguladores são tratados como transformadores de relação fixa $a = U_1/U_2$, e todos os capacitores como impedâncias *shunt* desligadas. No `ckt7`, a `RegulatingControl` atua apenas sobre dois `LinearShuntCompensator`, dado que a inspeção do `ckt7_CIM100x.XML` não localiza instâncias de `RatioTapChanger`, de modo que a tensão do circuito é pouco sensível a essa simplificação. No `ckt24`, que dispõe de um regulador de tensão sob carga (`RatioTapChanger`) na subestação e de três bancos de capacitores, esses controles, somados à modelagem da fonte, respondem pela diferença de tensão em relação ao `OpenDSS` (seção 5.3): o simulador opera o regulador e os capacitores para elevar a tensão da rede, e o programa, que mantém a fonte em tensão nominal e não modela esses controles, fica abaixo.

(b) Vector group não modelado. O atributo `connectionKind` dos `PowerTransformerEnd` (e equivalentes do caminho B) carrega a informação de conexão Δ -Y e o deslocamento angular associado. O programa, neste estágio, considera apenas o módulo da relação de transformação, descartando o deslocamento de fase de 30° característico das ligações Δ -Y. O efeito desse descarte manifesta-se no desvio sistemático de -35° observado nos ângulos da fase A, conforme discutido em 5.2.8.

(c) Efeito *shunt* desativado. A admitância *shunt* das linhas (representada por Y_m) e o ramo magnetizante dos transformadores (a partir dos atributos `excitingCurrent` e `loss` do `NoLoadTest`) são derivados pela rotina de tradução, mas mantidos desligados durante a varredura por motivos de estabilidade numérica (`Program.cs:L941`). A reativação exigiria modificação no método `calculatePF` para introduzir a fonte de corrente associada à derivação *shunt* a cada iteração.

(d) Critério de convergência exclusivamente em tensão. O critério adotado, $\max|\Delta V| < \text{tol}_V$, é numericamente estável e fisicamente interpretável, mas não fornece, por si só, verificação do balanço de potência ativo e reativo. A grandeza $|\Delta S|$ continua sendo gravada para inspeção, sem porém compor o teste de parada. Implementações futuras poderiam combinar os dois critérios, com terminação por ΔV e validação posterior do resíduo ΔS .

(e) Inicialização da varredura em nominalVoltage e impedância Thévenin do EnergySource não modelada. O parser CIM lê os atributos `voltageMagnitude`, `voltageAngle`, `r`, `x`, `r0` e `x0` do `EnergySource` (`Program.cs` linhas 1573 a 1579), mas o motor de

cálculo inicializa a raiz da varredura apenas com o `nominalVoltage` (Program.cs linha 440). Nos três circuitos do EPRI, o `voltageMagnitude` é declarado em 1,05 pu sobre V_{nominal} , com valor absoluto de 120,75 kV no ckt7 e 241,5 kV no ckt24, e a impedância Thévenin equivalente, embora explícita no CIM ($r = 2,314 \Omega$ e $x = 28,52 \Omega$ no ckt7; $r = 0,63 \Omega$ e $x = 6,72 \Omega$ no ckt24), não compõe ramo da árvore radial processada pelo programa. Como o terminal do `vSource` do OpenDSS é rebaixado pela própria impedância sob carga, enquanto o programa fixa a fonte na tensão nominal, essa escolha de modelagem responde pelo pequeno resíduo de tensão de $\approx +1\%$ observado no primário do ckt7 (seção 5.2.8), maior na barra de 115 kV da fonte (+3,16%) e atenuado ao longo do alimentador. No ckt24, a mesma escolha atua em sentido oposto: como a fonte do simulador é rígida e o regulador da subestação eleva o barramento regulado, fixar a fonte em 1,0 pu coloca o programa cerca de 3,7% abaixo do OpenDSS já na cabeça do alimentador (seção 5.3).

A partir das limitações expostas anteriormente, identificam-se seis direções de continuidade.

A primeira consiste em modelar o `EnergySource` como ramo equivalente Thévenin da rede a montante, aproveitando os atributos `voltageMagnitude`, `voltageAngle`, r , x , r_0 e x_0 já lidos pelo parser CIM (Program.cs linhas 1573 a 1579) e atualmente descartados pelo motor de cálculo. A adaptação envolve a inicialização da raiz da varredura em $V_{\text{magnitude}}$, e não mais em V_{nominal} , e a inserção de um ramo fictício de impedância Thévenin entre o nó interno da fonte e a barra `SOURCEBUS` da árvore radial. Com isso, o terminal da fonte passaria a refletir o afundamento sob carga reproduzido pelo OpenDSS, eliminando o pequeno resíduo de tensão discutido em 5.2.8.

A segunda direção consiste em modelar a classe `RegulatingControl` do CIM, com leitura dos atributos `targetValue`, `targetDeadband` e `monitoredPhase`, e implementação de laço externo que ajuste o estado dos bancos de capacitores chaveados e os *taps* dos reguladores de tensão em busca do ponto de regulação. Essa adição é o que falta para reproduzir a tensão regulada de circuitos como o ckt24, em que a diferença em relação ao OpenDSS decorre da combinação da fonte mantida em tensão nominal e da regulação ativa não modelada, isto é, o regulador da subestação e os capacitores (seção 5.3).

A terceira direção é a incorporação do *vector group* dos transformadores. A adaptação envolve a leitura do `connectionKind` na etapa de tradução e a aplicação do

deslocamento angular correspondente nas matrizes de transformação do bloco $\bar{C}_i$ da equação (2.1). A operação tem natureza estritamente sintática (não modifica o pseudocódigo da varredura) e exige apenas a inclusão de um deslocamento de fase no momento da reflexão $V_{para} \leftarrow V_{de}/a$.

A quarta direção corresponde à paralelização da varredura *backward/forward*, conforme proposto por Souza (2023) e De Souza, Leite e Reiz (2024). A estrutura de árvore radial admite particionamento natural por sub-ramais, e a literatura reporta ganhos de até 33% em quatro núcleos. A migração da DFS recursiva para versão *fork-join* deve aproveitar a biblioteca `Task Parallel Library` do .NET, sem alteração da estrutura de classes.

A quinta direção é o estudo de cenários horários ao longo de 8760 horas anuais, conforme adotado por Dugan e McDermott (2011) no OpenDSS, aproveitando a estrutura CIM para múltiplos perfis de carga e geração. A extensão exige principalmente a leitura das classes `LoadShape` e `GeneratorShape` do CIM, e a parametrização do laço externo do programa.

A sexta e última direção é a extensão para circuitos não radiais. O método BFS, na formulação adotada, pressupõe topologia em árvore. Alimentadores em malha, comuns em centros urbanos densos, exigem ou a abertura da malha em um ponto de injeção fictício, com posterior superposição de correntes circulantes, ou a substituição do método BFS por uma técnica iterativa de injeção de corrente compatível com topologia geral. Ambas as alternativas estão documentadas em Kersting (2002) e podem ser incorporadas ao programa em desenvolvimento futuro sem alteração das camadas de leitura do CIM e de tradução para a hierarquia Rede.

REFERÊNCIAS

BARROS, J. V.; LEITE, J. B. Development of a relational database oriented on the Common Information Model for power distribution networks. In: **2021 IEEE URUCON**. Montevidéu: IEEE, 2021. p. 63–66. DOI: 10.1109/URUCON53396.2021.9647155.

DE SOUZA, M.; LEITE, J. B.; REIZ, C. Efficient power flow algorithm for unbalanced three-phase distribution networks using recursion and parallel programming. In: **Mediterranean Electrotechnical Conference (MELECON)**, 2024.

DUGAN, R. C.; McDERMOTT, T. E. An open source platform for collaborating on smart grid research. In: **IEEE Power and Energy Society General Meeting**. Detroit: IEEE, 2011. p. 1–7.

DUGAN, R. C.; MONTENEGRO, D. **The Open Distribution System Simulator (OpenDSS): Reference Guide**. Palo Alto: Electric Power Research Institute, 2020.

ELECTRIC POWER RESEARCH INSTITUTE. **EPRI Test Circuits**. GitHub: dss-extensions/electricdss-tst, [s.d.]a. Disponível em: <https://github.com/dss-extensions/electricdss-tst>. Acesso em: 27 maio 2026.

ELECTRIC POWER RESEARCH INSTITUTE. **Options**. OpenDSS Documentation. Palo Alto: EPRI, [s.d.]b. Disponível em: <https://opendss.epri.com/Options.html>. Acesso em: 27 maio 2026.

ELECTRIC POWER RESEARCH INSTITUTE. **Summary of EPRI Test Circuits**. Palo Alto: EPRI, [s.d.]c. Disponível em: <https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/Distrib/EPRI/TestCircuits/>. Acesso em: 16 maio 2026.

FERREIRA, D. S. **Análise do Algoritmo de Fluxo de Potência e da Extensão do Modelo de Informação Comum para Redes de Distribuição de Energia Elétrica**. 2021. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Faculdade de Engenharia de Ilha Solteira, Universidade Estadual Paulista, Ilha Solteira, 2021.

GEROLOMETTO, N. **Análise de desempenho dos algoritmos de cálculo de fluxo de potência em redes de distribuição de energia elétrica com geração distribuída**. 2023. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Faculdade de Engenharia de Ilha Solteira, Universidade Estadual Paulista, Ilha Solteira, 2023.

INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC 61968-11: Application integration at electric utilities – System interfaces for distribution**

management – Part 11: Common Information Model (CIM) extensions for distribution. Genebra: IEC, 2013.

INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC 61970-301: Energy management system application program interface (EMS-API) – Part 301: Common Information Model (CIM) Base.** Genebra: IEC, 2020.

KERSTING, W. H. **Distribution System Modeling and Analysis.** 2. ed. Boca Raton: CRC Press, 2002.

LEITE, J. B.; MANTOVANI, J. R. S. Development of a smart grid simulation environment, Part I: Project of the electrical devices simulator. **Journal of Control, Automation and Electrical Systems**, v. 26, n. 1, p. 80–95, fev. 2015.

SOUZA, M. **Desenvolvimento de um algoritmo eficiente de cálculo do fluxo de potência em redes de distribuição de energia elétrica usando recursão e programação paralela.** 2023. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Faculdade de Engenharia de Ilha Solteira, Universidade Estadual Paulista, Ilha Solteira, 2023.