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

Israel Panazollo

METODOLOGIA PARA MODELAGEM E ANÁLISE INTEGRADA DE REDES DE TRANSMISSÃO E DISTRIBUIÇÃO NO SOFTWARE OPENDSS

Santa Maria, RS
2022

Israel Panazollo

METODOLOGIA PARA MODELAGEM E ANÁLISE INTEGRADA DE REDES DE TRANSMISSÃO E DISTRIBUIÇÃO NO SOFTWARE OPENDSS

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica, Universidade Federal de Santa Maria (UFSM, RS), como requisito para obtenção do título de **Engenheiro Eletricista.** Defesa realizada por videoconferência.

ORIENTADOR: Prof. Daniel Pinheiro Bernardon

Santa Maria, RS
2022

End. Eletr.: israelpanazollo@gmail.com

Israel Panazollo

METODOLOGIA PARA MODELAGEM E ANÁLISE INTEGRADA DE REDES DE TRANSMISSÃO E DISTRIBUIÇÃO NO SOFTWARE OPENDSS

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

Aprovado em 30 de junho de 2022:

_____

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

_____

Émerson Rafael da Silva, Me. (UFSM) (videoconferência)

_____

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

Santa Maria, RS
2022

AGRADECIMENTOS

Agradeço a Deus, criador e sustentador da realidade, pelo dom da vida e o dom da razão para discernir os fenômenos do cosmo, a Jesus Cristo pela vida eterna obtida pelo seu precioso sacrifício e concedida a todos os que creem e ao Santo Espírito por iluminar, consolar e guiar na jornada.

Agradeço a minha família pelas lições ensinadas, pelos bons valores transmitidos, pelo cuidado e apoio para chegar até aqui. A meu pai, Maximiliano, pelo exemplo de trabalha duro e altruísmo, a minha mãe (in memorian), Silvana, pelas orações e todo carinho e aos meus irmãos Ramon e Ábner, por me ajudarem a ser um irmão melhor e divertirem a minha vida.

Agradeço a minha querida amiga e companheira Schadia Abdel, por todo amor, suporte e por alegrar a minha caminhada nesse mundo.

Agradeço aos meus amigos, com os quais troco aprendizados, compartilho os fardos, curto a vida e sirvo no Reino de Deus. Destaco aqui Bruno Chaves, Esdras Tonin, Ismael Rosa, Jeilson Nascimento, Lucas Dowsley e a todos do Farol.

Agradeço aos meus colegas de curso, que me auxiliaram nas dúvidas, facilitaram o trabalho e sofreram junto os desafios do curso. Em especial ressalto aqueles que contribuíram especificamente na elaboração deste trabalho, como a Viviane Lima e o Gustavo Cordeiro.

Por fim, agradeço ao professor Daniel Pinheiro Bernardon por toda ajuda, instrução, compreensão, prontidão em responder e dicas para solucionar os problemas. Agradeço também ao professor Maurício Sperandio pelo auxílio valioso e material compartilhado que possibilitou o estudo de um caso real.

RESUMO

METODOLOGIA PARA MODELAGEM E ANÁLISE INTEGRADA DE REDES DE TRANSMISSÃO E DISTRIBUIÇÃO NO SOFTWARE OPENDSS

AUTOR: Israel Panazollo
ORIENTADOR: Daniel Pinheiro Bernardon

Este trabalho propõe desenvolver uma metodologia na qual seja possível modelar e analisar de maneira integrada as redes de transmissão e de distribuição. Uma vez que há várias ferramentas computacionais com diferentes formas de operação, funcionalidades e aplicações, as empresas ou organizações podem empregar diferentes programas. Em se tratando de análises no âmbito da transmissão de energia, pode-se utilizar o Programa de Análise de Redes (ANAREDE), enquanto para o contexto distribuição da energia, é possível empregar o *Open Distribution System Simulator* (OpenDSS). Para realizar a integração das redes aplicou-se a linguagem Python, de modo que em apenas um software, o OpenDSS, sejam realizados as análises. Uma rotina em Python foi elaborada na qual se convertem os cartões provenientes do ANAREDE e colocam os dados dentro de um ambiente em Python. Posteriormente outro algoritmo criado, gerará um arquivo, que representa a conversão da rede modelada do ANAREDE, que será lida pelo OpenDSS. Aplicou-se este processo a dois exemplos do Centro de Pesquisa de Energia Elétrica (CEPEL), encontrando resultados satisfatórios de fluxo de potência comparando os dois programas. Por fim, uma rede de transmissão real, Litoral Norte, é convertida do ANAREDE para OpenDSS e esta é conectada com um alimentador da rede de distribuição de Tramandaí, modelado no OpenDSS. Assim verificam-se os impactos da rede de distribuição na rede de transmissão e encontram-se valores razoáveis com respeito ao fluxo de potência, mesmo testando em alguns cenários distintas.

Palavras-chave: OpenDSS. ANAREDE. Phyton. Integração de Redes. Transmissão de Energia. Distribuição de Energia.

ABSTRACT

METHODOLOGY FOR MODELING AND INTEGRATED ANALYSIS OF TRANSMISSION AND DISTRIBUTION NETWORKS IN OPENDSS SOFTWARE

AUTHOR: Israel Panazollo
ADVISOR: Daniel Pinheiro Bernardon

This work intend to develop a methodology in which would be possible to model and analyze in a integrated way transmission and distribution grids. Since there are plenty computational tools with different modes of operation, functionalities and applications, companies and organizations can use various programs. With respect to the analyzes of energy transmission scope, the *Programa de Análise de Redes* (ANAREDE) could be use, while to the context of energy distribution, it is possible to adopt the Open Distribution System Simulator (OpenDSS). In order to engender grid integration Pyhton language was applied, so that in only software, OpenDSS, the analyzes were performed. A Python routine was elaborate in which the cards from ANAREDE were converted and the data was stored in a Python environment. Posteriorly another developed algorithm will generate a file that represents the conversion of the modeled grid of ANAREDE, which will be read by OpenDSS. It were applied this process in two examples of *Centro de Pesquisa de Energia Elétrica* (CEPEL), finding satisfactory results of power flow comparing both programms. Finally, a real transmission grid, *Litoral Norte*, were converted from ANAREDE to OpenDSS and were connected with a feeder from *Tramandaí* distribution grid, modeled in OpenDSS. Thus, impacts of the distribution grid in the transmission grid were verified and reasonable values with respect to the power flow were found, even with distinct scenarios.

Keywords: OpenDSS. ANAREDE. Phyton. Grid Integration. Energy Transmission. Energy Distribution.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

$ANEEL$	Agência Nacional de Energia Elétrica
$ANAREDE$	Programa de Analise de redes
CA	Corrente Alternada
CC	Corrente Contínua
$CEPEL$	Centro de Pesquisa de Energia Elétrica
COM	Component Object Model
$DBAR$	Código de execução para leitura dos dados de barra CA
$DGBT$	Código de execução para leitura dos grupo base de tensão
$DLIN$	Código de execução para leitura dos dados de circuito CA
$DCTE$	Código de execução para leitura/modificação de dados de constantes.
DLL	*Dynamic Link Library*
FP	Fator de Potência
GD	Geração Distribuída
LTC	Variação automática de tap
$MUST$	Montantes de Uso de Sistema de Transmissão
ONS	Operador Nacional do Sistema
$OpenDSS$	Open Distribution System Simulator
PC	*Power Conversion*
PD	*Power Delivery*
SEP	Sistema Elétrico de Potência
SIN	Sistema Interligado Nacional
tap	*Transformer Adjust Position*

LISTA DE SÍMBOLOS

A Ampères

cos cosseno

e Número de euler

Hz Hertz

j Número complexo

k kilo, 10^3

M Mega, 10^6

$p.u.$ Valor por unidade

sen seno

V Volts

VA Volt-Ampère

var Volt-Ampère reativo

W Watts

Δ Letra grega delta

θ Letra grega theta

Ω Letra grega ômega

$|x|$ Valor em módulo

$\overrightarrow{x}$ Valor fasorial

SUMÁRIO

1 INTRODUÇÃO

Para a *National Academy of Engineering*, os sistemas elétricos de potência são uma maravilha técnica, sendo uma das maiores conquistas da engenharia do século XX (WULF W. M. A., 2000). No mundo moderno, é difícil imaginar uma sociedade sem energia elétrica e sem as facilidades derivadas da eletricidade (MOHAN, 2016).

No fim do século XIX temos o surgimento das linhas de transmissão de energia elétrica, com o objetivo, num primeiro momento, de suprir os sistemas de iluminação e mais tarde, com as demandas de motores elétricos, surgem as companhias de luz e força (OLIVEIRA et al., 2000). O registro mais antigo de uma central de energia elétrica data de 1881 na Inglaterra, onde duas rodas d'água geravam uma corrente alternada usada para abastecer lâmpadas (PINTO, 2018).

No Brasil, a primeira linha de transmissão foi construída em Diamantina, em torno de 1883, sendo uma linha de 2 km, ligada a uma pequena usina hidrelétrica (PINTO, 2018). No país, as primeiras usinas e linhas de transmissão tinham como finalidade alimentar cargas pontuais, contudo, o setor elétrico brasileiro foi crescendo e hoje adquiriu proporções imensas, procurando cumprir a missão de suprir energeticamente tanto os grandes centros urbanos como as regiões mais remotas do Brasil (BARROS; BORELLI; GEDRA, 2014).

As redes elétricas brasileiras, agora com mais de um século de existência, que compõem o Sistema Interligado Nacional (SIN), estão dividas em 4 subsistemas: Norte, Nordeste, Sudeste / Centro-Oeste e Sul (ROBBA et al., 2020). O sistema elétrico do Brasil apresenta um elevado nível de complexidade e ramificações, uma vez que se estende por um país de tamanho continental, formando um sistema que abrange quase todas as regiões do país e as interliga (BARROS; BORELLI; GEDRA, 2014). Há 212 localidades no Brasil que não estão integradas sendo a maior parte da região Norte e representando cerca de 1% da carga do país (ONS, 2022). Para Robba et al. (2020) as demandas energéticas podem ser atendidas adequadamente com as malhas de redes de transmissão e com a transferência de energia entre os subsistemas.

Devido a vastidão e complexidade que os sistemas elétricos podem atingir, a realização de diagnósticos, bem como análises e planejamento podem precisar de ferramentas avançadas que sejam capazes de processar e calcular com um grande volume de informações (MOHAN, 2016). Se levar em conta novos desafios, como a inserção de geração distribuída (GD) a tarefa da correta operação do sistema elétrico pode se elevar ainda mais (COSTER et al., 2011).

Para efetuar análise nas redes elétricas usam-se ferramentas de cálculo. Estas ferramentas experimentaram modificações e melhorias com o passar do tempo. Se no início, havia os diagramas de círculo, que possibilitavam a representação gráfica da variação de potência, tensão e corrente em uma rede, com o advento da computação as ferramentas

foram sendo implementadas em computadores, seja com o método de Gauss-Seidel em computadores à válvula, na década de 50, ou com o método de Newton-Raphson em computadores com transistores, em meados de 60, até hoje com equipamentos e programas avançados que processam um grande volume de dados em um tempo reduzido (ROBBA et al., 2020).

O Programa de Análise de Redes (ANAREDE) é uma das ferramentas que mais se destacam no Brasil no campo dos sistemas elétricos de potência (SEP), sendo uma ferramenta computacional desenvolvida pelo Centro de Pesquisa de Energia Elétrica (CEPEL), que inclui funcionalidades que possibilitam encontrar equivalentes de rede e resolver problemas de fluxo de potência, sensibilidade, contingências e segurança(ROBBA et al., 2020).

O *Open Distribution System Simulator* (OpenDSS) é outro programa que tem ganho destaque no âmbito da análise e planejamento para a rede de distribuição, tendo como diferencial, permitir a elaboração de redes fragmentadas e articulação das diferentes partem em uma simulação, sendo esta uma característica útil devida as alterações constantes no setor de distribuição (ANDRADE et al., 2020).

1.1 MOTIVAÇÃO E JUSTIFICATIVA

As concessionárias de energia, bem como organizações e empresas possuem diversas opções de ferramentas computacionais a disposição para realizarem diagnósticos, planejamentos e análises das redes elétricas. Quando se foca no setor da transmissão de energia, o Operador Nacional do Sistema (ONS), indica no Submódulo 18.2 que o software de referência de análise de redes é o ANAREDE (ONS, 2008). Para o ramo da distribuição de energia, um software que tem sido usado amplamente, tanto na indústria, quanto na academia é o OpenDSS (SOUZA; FERREIRA; CAMPOS, 2020).

Levando em conta esses diferentes programas, o desafio seria analisar de maneira integrada esses dois setores das redes elétricas em um programa ou ambiente. Desse modo seria poderiam ser desempenhadas análises que levam em conta o impacto das redes de transmissão nas redes de distribuição e vice-versa, ou seja, análises mais abrangentes. Destacam aqui alguns resultados de interesse, como o fluxo de potências, limites de tensão, perdas, entre outros valores relevantes, podendo propiciar um diagnóstico e planejamento mais amplo.

1.2 OBJETIVOS GERAL

Este trabalho tem como objetivo principal desenvolver uma metodologia que permita realizar a análise integrada de redes de transmissão e distribuição utilizando o software OpenDSS. Para tanto será preciso utilizar programação em Python, além de conhecer melhor os programas ANAREDE e OpenDSS.

1.3 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

1. Entender e analisar melhor o comportamento de uma rede elétrica, seus elementos e valores;
2. Compreender o funcionamento do software ANAREDE, do qual virão os cartões com os dados da transmissão;
3. Estudar sobre como software OpenDSS opera, uma vez que no final haverá um cartão com informações convertidas para um formato que este software lê e processa;
4. Aprofundar os conhecimentos sobre programação e utilização da linguagem Python, dado que será utilizado um ambiente nessa linguagem de programação para realizar a conversão;
5. Desenvolver um método de integração das redes no OpenDSS;
6. Aplicar a metodologia em um caso e analisar os resultados;
7. Possibilitar a formulação de estratégias de correção e melhoria da rede ao analisar de maneira integrada.

1.4 ORGANIZAÇÃO DOS CAPÍTULOS

O trabalho está dividido em 5 capítulos. No primeiro capítulo há a introdução, com as considerações iniciais sobre o tema, bem como objetivos gerais, específicos e motivação para o trabalho. No segundo capítulo, é apresentado o referencial teórico no qual se aborda a análise de redes, o software ANAREDE e OpenDSS e a linguagem Python. No terceiro capítulo, trata-se da metodologia em si, que envolve a conversão do cartão do ANAREDE para um cartão em OpenDSS no ambiente Python e, posteriormente, a

proposta de integração no OpenDSS. No quarto capítulo, são analisados alguns casos utilizando a ferramenta computacional, primeiro validando a conversão de dois cartões do ANAREDE no OpenDSS e então testando em uma rede real. Por fim, no quinto capítulo são apontadas algumas considerações finais sobre o trabalho e o tema. Os anexos incluem os códigos e alguns resultados de interesse.

2 REFERENCIAL TEÓRICO

Neste capítulo será discorrido sobre os principais conceitos e tópicos que servirão de fundamento para compreender a temática do trabalho. Primeiramente, tratar-se-á das rede de energia elétrica, seus elementos e como analisar o seu comportamento. Então, serão abordados os softwares ANAREDE e OpenDSS, enfatizando a forma de modelar os elementos da rede nestes softwares. Por fim, será abordado brevemente sobre a linguagem Python e alguns comandos relevantes para o desenvolvimento dos algoritmos computacionais.

2.1 ANÁLISE DE REDES

2.1.1 Grandezas elétricas

Ao tratar da energia elétrica é preciso definir e compreender algumas grandezas. A corrente elétrica é entendida como a quantidade de cargas elétricas que fluem por um condutor num dado intervalo de tempo, dado em Ampères (A). A tensão elétrica é a diferença de potencial elétrico, dado em Volts (V). A potência elétrica é a quantidade de energia que flui num intervalo de tempo (HALLIDAY; RESNICK; WALKER, 2020).

$$i = \frac{dq}{dt} \tag{2.1}$$

2.1.2 Transmissão CC ou CA

A energia elétrica pode ser transmitida em corrente contínua (CC) ou em corrente alternada (CA). No Brasil, a transmissão em CA é a forma mais comum no Brasil (PINTO, 2018). Os sistemas elétricos podem variar com relação ao número de fases, seja monofásico ou polifásicos, muito embora no Brasil o tipo de SEP mais frequente é trifásico (OLIVEIRA et al., 2000). As grandezas elétricas serão tratadas como fasores, ou seja, possuem módulo e ângulo, uma vez que o comportamento senoidal em CA, exige isso. As Equações 2.2 e 2.3, indicam os fasores de tensão e corrente, respectivamente (PINTO, 2018).

$$\overrightarrow{V} = |V|.\angle\theta_V \tag{2.2}$$

$$\vec{I} = |I|.\angle\theta_I \tag{2.3}$$

A tensão e a corrente dadas como fasores podem ser convertidas em sua forma polar. Uma tensão V com ângulo θ, por exemplo, será escrita conforme a Equação 2.4, inclusive podendo ser rescrita conforme a Equação 2.5, para explicitar o comportamento senoidal.

$$\vec{V} = |V|.e^{j.\theta} \tag{2.4}$$

$$\vec{V} = |V|.(cos\theta + j.sen\theta) \tag{2.5}$$

Em se tratando de transmissão CA, haverá uma frequência de trabalho, que no Brasil é 60 Hz, sendo a mesma frequência de país como Estados Unidos, Canadá, México, Equador e outros. Vale ressaltar que na Europa bem como em boa parte da África e Ásia a frequência é 50 Hz. A escolha de 60 Hz tem como vantagem permitir equipamentos do sistema menores, muito embora as reatâncias das linhas de transmissão são um pouco maiores quando comparadas a 50 Hz (PINTO, 2018).

2.1.3 Leis de Kirchhoff

Os circuitos elétricos estão sujeitos a lei de conservação de carga, isto é, que cargas não podem ser criadas, nem destruídas em um nó, bem como a lei de conservação de energia, a saber, que a energia num sistema vai sendo convertida em outras formas, não podendo ser gerada nem eliminada do nada. Dessas leis encontram-se a lei das tensões Kirchhoff, que diz que "a soma algébrica das variações de potencial encontradas ao longo de uma malha completa de um circuito é zero"e a lei das correntes Kirchhoff que aponta que "a soma das corrente que entram em um nó é igual à soma das correntes que saem do nó" (HALLIDAY; RESNICK; WALKER, 2020).

2.1.4 Impedância

Os equipamentos da rede possuem componentes com diferentes grandezas. Estes podem possuir resistências e reatâncias que são valores associados a oposição da passagem de corrente elétrica. A impedância (Z) é a soma da resistência (R) e da reatância (X) de um elemento na rede, como indicado na Equação 2.6. Matematicamente, a resistência corresponde a parte real, enquanto a reatância está associada a parte imaginária (PINTO, 2018). Além disso, outro conceito importante é a admitância, definida como o inverso da

impedância, mostrada na Equação 2.7 (MOHAN, 2016).

$$Z = R + j.X \tag{2.6}$$

$$Y = \frac{1}{Z} \tag{2.7}$$

2.1.5 Potência complexa

A potência complexa (S) possui uma parte real e outra imaginária, dada em V.A, mostrada na Equação 2.8. A potência ativa, medida em Watts (W), representa a parte efetivamente produz trabalho. A potência reativa, com valores em var, é usada para manter o campo eletromagnético e não produz trabalho útil. Além disso, outro valor importante é o fator de potência, apresentado na Equação 2.9 que relaciona a potência ativa e potência aparente (PINTO, 2018).

$$S = P + j.Q \tag{2.8}$$

$$FP = \frac{P}{S} \tag{2.9}$$

2.1.6 Valor por unidade

Um recurso matemático facilita consideravelmente o cálculo de redes, em especial quando há transformadores: os valores por unidade (p.u.) que correspondem a uma mudança de escala das grandezas (OLIVEIRA et al., 2000). Haverá valores de base, que são definidos, e os valores reais, de modo que as grandezas elétricas serão convertidas conforme a Equação 2.10 nos valores em p.u. Dentre as grandezas há a tensão, a corrente, a potência e a impedância.

$$Valor_{pu} = \frac{Valor_{real}}{Valor_{base}} \tag{2.10}$$

2.1.7 Sistema elétrico de potência

A função básica de um sistema elétrico de potência é suprir os consumidores com energia elétrica na quantidade solicitada, com qualidade adequada, no momento que for

demandado. Um SEP é composto fundamentalmente de três blocos, a saber, a geração, a transmissão e a distribuição. A geração de energia diz respeito a fontes que produzem energia elétrica, ou melhor, transformam um tipo de energia em outro. A transmissão de energia está associada a transferência de potência elétrica dos locais de geração até grandes centros que recebem esta energia. A distribuição de energia, por sua vez, trata do transporte dessa energia dos pontos de entrega até os consumidores finais (ROBBA et al., 2020).

2.1.8 Tipos de barras

Os SEPs são compostos de diversos equipamentos com características distintas. As linhas conectam os barramentos e possuem valores de impedância, que dependem do material, dimensões e comprimento. Na rede podem estar conectados transformadores, geradores, motores, banco de capacitores, entre outros. As barras recebem a sua classificação conforme a sua função, havendo as barras PQ ou de carga, as barras PV ou de tensão controlada e as barras $V\theta$ ou *swing* ou ainda de referência. Nas barras do tipo PQ se conhecem os valores das potências ativas e reativas, enquanto nas barras PV sabe-se o valor da potência ativa e do módulo da tensão, e, por fim, nas barras $V\theta$, tem-se o módulo e o ângulo da tensão conhecidos (ROBBA et al., 2020).

2.1.9 Modelo de Carga

As cargas em um sistema elétrico podem ser das mais variadas, por isso as concessionárias podem representá-las de modos diferentes, de acordo com o comportamento da carga, de modo a tratá-las ou como carga de impedância constante ou potência constante ou ainda corrente constante (MOHAN, 2016).

2.1.10 Fluxo de potência

Para planejamento de redes elétricas é fundamental conhecer a capacidade de transferência de potência através das linhas para atender as cargas (MOHAN, 2016). A análise do fluxo de carga ou fluxo de potência é essencial também para compreender o comportamento de um sistema elétrico de potência (SILVEIRA et al., 2021). O estudo do fluxo de potência pode ser entendido como uma simulação em regime permanente da operação de um sistema elétrico com certas grandezas com graus de liberdade onde se determina através de equações as outras grandezas (ROBBA et al., 2020).

Um sistema elétrico de potência está sujeito as leis de Kirchhoff de modo que se considera a conservação de potência em cada nó e barra do sistema (MONTICELLI, 1983 apud SILVEIRA, 2021). A injeção de corrente em um barramento k se relaciona com a tensão e as impedâncias e admitância (MOHAN, 2016). Pela lei das correntes de Kirchhoff, obtem-se a Equação 2.11, onde $\vec{V_k}$ é o fasor da tensão na barra, $\vec{V_m}$ o fasor da tensão em outra barra, Y_{kG} é a soma de admitância conectadas no barramento k ao terra, Z_{km} a impedância entre os barramentos ou impedância série e Y_{kk} a admitância própria ou autoadmitância. Com isso é possível obter a matriz de admitâncias dos barramentos, Y_{bus}, e rescrever a injeção de corrente num sistema de n barramentos através da equação matricial.

$$\vec{I_k} = \vec{V_k}.Y_{kG} + \sum_{m \neq k} \frac{\vec{V_k} - \vec{V_m}}{Z_{km}} \tag{2.11}$$

$$Y_{km} = -\frac{1}{Z_{km}} \tag{2.12}$$

$$Y_{kk} = Y_{kG} + \sum_{m \neq k} \frac{1}{Z_{km}} \tag{2.13}$$

$$\begin{bmatrix} \vec{I_1} \\ \vec{I_2} \\ ... \\ \vec{I_n} \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & ... & Y_{1n} \\ Y_{21} & Y_{22} & ... & Y_{2n} \\ ... & ... & ... & ... \\ Y_{n1} & Y_{n2} & ... & Y_{nn} \end{bmatrix} . \begin{bmatrix} \vec{V_1} \\ \vec{V_2} \\ ... \\ \vec{V_n} \end{bmatrix} \tag{2.14}$$

A potência complexa injetada na barra pode ser calculada conforme a Equação 2.15, isto é, multiplicando o fasor da tensão na barra, V_k, pelo fasor da corrente na barra k, I_k, conjugado, resultando em uma componente de potência ativa, P_k, e uma componente reativa, Q_k (MOHAN, 2016). Definindo G, como o inverso da resistência, ou seja, a condutância e designando B, como o inverso da reatância, isto é, a susceptância, obtém-se a Equação 2.16 (STEVENSON, 1986 apud SILVEIRA, 2021).

$$\vec{S_k} = P_k + j.Q_k = \vec{V_k}.\vec{I_k^*} \tag{2.15}$$

$$Y_{km} = G_{km} + j.B_{km} \tag{2.16}$$

Ao escrever na forma de admitâncias, colocar na forma polar as tensões e separar a parte real e imaginária, o resultado do cálculo da potência complexa injetada na barra serão as Equações 2.17 e 2.18 (MOHAN, 2016). Vale destacar que $\theta_{km} = \theta_k - \theta_m$ e que

$cos\theta_{kk} = 1$ e $sen\theta_{kk} = 0$.

$$P_k = G_{kk}.|V_k|^2 + |V_k|.\sum_{m=1;m\neq k}^{n} |V_m|.(G_{km}.cos\theta_{km} + B_{km}.sen\theta_{km}) \tag{2.17}$$

$$Q_k = -B_{kk}.|V_k|^2 + |V_k|.\sum_{m=1;m\neq k}^{n} |V_m|.(G_{km}.cos\theta_{km} - B_{km}.sen\theta_{km}) \tag{2.18}$$

2.1.11 Método de Newton-Raphson

O método de Newton-Raphson foi introduzido na década de 1960 para solucionar o problema de fluxo de potência (TINEEY;HART,1967 apud ROBBA et al.,2020). Desde então tem sido o método preferido dada robustez elevada e resolução de sistemas de equação não lineares com boa convergência (ROBBA et al., 2020). Para utilizar o método é preciso modelar a rede através da sua matriz de admitância e à partir das equações básicas de fluxo de potência realizar o procedimento de Newton-Raphson (MOHAN, 2016).

Sendo um método iterativo, o objetivo é encontrar um valor de x que satisfaça a Equação 2.19, onde $f(x)$ é uma função não linear e c é uma constante. Para tanto, parte-se de uma estimativa inicial, $x^{(0)}$. Realizam-se ajustes pequenos, Δx, à fim de que a solução chegue mais perto do real, com base na Equação 2.21, que é estimado para reduzir o erro, ϵ, abaixo de uma tolerância especificada, apresentado Equação 2.22 . A cada nova iteração terá a Equação 2.20 (MOHAN, 2016).

$$c - f(x) = 0 \tag{2.19}$$

$$x^{(1)} = x^{(0)} + \Delta x \tag{2.20}$$

$$\Delta x = \frac{c - f(x^{(0)})}{\frac{\partial f}{\partial x}} \tag{2.21}$$

$$\epsilon = |c - f(x)| \tag{2.22}$$

Para a aplicação do método de Newton-Raphson no fluxo de carga, levará em conta as Equações 2.17 e 2.18 na forma matricial, apresentada na Equação 2.23. Dependendo do número de barras e do tipo delas se estipulará o tamanho das matrizes. Para as barras PQ se calculará o módulo da tensão e o ângulo, para as barras PV, busca-se o valor do ângulo, já que o módulo da tensão já é conhecido. O procedimento envolverá uma matriz

de derivadas parciais, uma matriz Jacobiana, que será composta de submatrizes.

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial \vec{V}} \\ \frac{\partial Q}{\partial \theta} & \frac{\partial Q}{\partial \vec{V}} \end{bmatrix} \cdot \begin{bmatrix} \Delta \theta \\ \Delta \vec{V} \end{bmatrix} \tag{2.23}$$

2.2 ANAREDE

O Programa de Análise de Redes, ANAREDE, é constituído de um conjunto de aplicações computacionais integradas. O software foi criado pelo Centro de Pesquisas de Energia Elétrica (CEPEL) e busca fornecer ao setor elétrico, subsídios para a realização de estudos nas redes elétricas, seja para o planejamento ou ainda operação de sistema elétricos de potência (ANAREDE, 2019).

A versão V11.00.01 inclui programas que realizam o fluxo de potência, análise de contingências, equivalentes de rede, análises de sensibilidade para tensão ou fluxo, dentre outras funcionalidades. O programa está acessível para computadores com sistema operacional Windows em versão 2000 ou superior. Os dados de entradas estão num arquivo com extensão .pwf, definidos em códigos de execução, . Ademais, o ANAREDE permite que uma interface gráfica seja usada para visualizar a rede analisada, nesse caso os arquivos encontram-se numa extensão .lst (ANAREDE, 2019).

Uma das principais funcionalidades é executar fluxos de potência. Os dois métodos à disposição resolver as equações associadas aos SEPs são o método desacoplado rápido e o método de Newton. os cálculos terão como condições inicias os dados pelos valores de entrada ou ainda pela opção *flat* que coloca os valores padrão. Cabe ressaltar que um sistema CC pode ser solucionado, muito embora será enfatizado neste trabalho os cálculos envolvendo rede CA (ANAREDE, 2019).

A solução é considerada convergente se todas as equações do sistema CA estiverem convergidas. È válido ressaltar que limites de tensão nas barras, limites de geração de reativos, variação de tap nos transformadores e limites de carregamento nas linhas são aplicados e a solução deve respeitar estes limites (ANAREDE, 2019).

As barras CA é desenvolvida segundo o modelo completo do fluxo de potência. Para representar as linhas e os transformadores são modelados como uma reatância em série entre os terminais. Os geradores são considerados como potência ativa fixa com variação de potência reativa que visa manter a tensão constante. As cargas podem ser pensadas de acordo com o modelo ZIP, ou seja, impedância constante ou corrente constante ou então potência constante, podendo ser dada até uma certa proporção para os tipos de carga (ANAREDE, 2019).

O software trabalho com códigos de execução que reúnem diferentes dados do

sistema de potência. Os códigos focados nesta aplicação serão DBAR que apresenta os dados das barras CA, o DLIN que mostra os dados do circuito CA, o DGBT que indica os grupos de base de tensão de barras CA. Há outros códigos que são importantes, porém não serão utilizados nas conversões. Contudo, vale destacar o DCTE, que tem a função de leitura e modificação de dados constantes do programa. Os dados de entrada essenciais para realizar o fluxo de potência são associados a topologia da rede e ao carregamento do sistema. Cabe ressaltar que dados adicionais tanto para geradores, como cargas ou demais elementos, bem como de contingências e casos podem ser adicionados no cartão para análises mais complexas sejam desenvolvidas. Contudo, o enfoque deste trabalho será para casos mais básicos (ANAREDE, 2019).

O códigos de execução DBAR apresenta os dados das barras CA, sendo que no cartão, cada linha informa os dados de uma barra específica, enquanto as colunas são agrupadas em campos que indicarão esses dados por tipo. Os campos, bem como as colunas e seus valores padrão são apresentados na Tabela 2.1. É válido descrever os campos apresentados na Tabela 2.1 (ANAREDE, 2019).

Tabela 2.1 – Descrição dos dados em DBAR.

Campo	**Colunas**	**Default**
Número	01-05	
Operação	06-06	A
Estado	07-07	L
Tipo	08-08	0
Grupo de Base de Tensão	09-10	0
Nome	11-22	
Grupo de Limite de Tensão	23-24	0
Tensão	25-28	1.0
Ângulo	29-32	0.0
Geração Ativa	33-37	0.0
Geração Reativa	38-42	0.0
Geração Reativa Mínima	43-47	*
Geração Reativa Máxima	48-52	*
Barra Controlada	53-58	**
Carga Ativa	59-63	0.0
Carga Reativa	64-68	0.0
Capacitor Reator	69-73	0.0
Área	74-76	1
Tensão para definição de carga	77-80	1.0
Modo de visualização	81-81	0.0
Agregador 1-5	82-96	

Fonte: Adaptado de ANAREDE (2019).

- Número: Número de identificação da barra CA;

- Operação: A ou 0 para adição de dados de barra, E ou 1 para eliminação de dados de barra e M ou 2 para modificação de dados de barra;
- Estado: L se a barra estiver em operação (ligado) e D se a barra circuito estiver fora de operação (desligado);
- Tipo: 0 para barra de carga (PQ), 1 para barra de tensão regulada (PV), 2 para barra de referência ($V\theta$) e 3 para barra de carga com limite de tensão (PQ enquanto a magnitude de tensão permanecer entre os valores limites);
- Grupo de Base de Tensão: Identificador de Grupo Base de Tensão ao qual pertence a barra CA, composto por até dois caracteres do tipo dígito (0 a 9) ou carácter (A a Z), conforme definido no Código de Execução DGBT. Os valores associados aos Grupos Base de Tensão são definidos no código de execução DGBT. Os grupos que não forem definidos terão valor igual a 1 kV;
- Nome: Identificação alfanumérica da barra;
- Grupo de Limite de Tensão: Identificador de Grupo de Limite de Tensão ao qual pertence a barra CA, composto por até dois caracteres do tipo dígito (0 a 9) ou carácter (A a Z), conforme definido no Código de Execução DGLT. Os valores associados aos Grupos de Limite de Tensão são definidos no Código de Execução DGLT. Os grupos que não forem definidos terão valores limites de tensão, mínimo e máximo, iguais a 0.8 e 1.2 p.u., respectivamente;
- Tensão: Valor inicial da magnitude da tensão, em p.u. Para barra de tensão controlada, remotamente ou não, por geração de potência reativa ou por variação de tap de transformador, este campo deve ser preenchido com o valor da magnitude da tensão a ser mantido constante;
- Ãngulo: Ângulo de fase inicial da tensão da barra, em graus;
- Geração Ativa: Valor de geração de potência ativa na barra, em MW. Este campo define o ponto base de operação sobre o qual as ações de controle são executadas de modo a manter o intercâmbio de potência ativa programado entre áreas. Os erros de intercâmbio de potência ativa entre áreas são distribuídos entre os geradores das áreas, com base neste valor e de acordo com a participação de cada gerador;
- Geração Reativa: Valor de geração de potência reativa na barra, em Mvar. Para barra de carga este valor é fixo. Para barra de carga com limite de tensão este valor é mantido constante, enquanto a magnitude da tensão permanecer entre os limites especificados. Para barras de tensão regulada e de referência com limites de geração de potência reativa especificados, este campo pode ser deixado em branco;

- Geração Reativa Mínima: Valor do limite mínimo de geração de potência reativa na barra, em Mvar;
- Geração Reativa Máxima: Valor do limite máximo de geração de potência reativa na barra, em Mvar;
- Barra Controlada: Para barras de tensão regulada e de referência, com limites de potência reativa especificados, este campo destina-se ao número da barra cuja magnitude da tensão será controlada. O valor da magnitude da tensão a ser mantido é obtido no campo Tensão do registro relativo à barra;
- Carga Ativa: Valor da carga ativa da barra, em MW.No caso da carga variar com a magnitude da tensão da barra, entre neste campo o valor da carga para a tensão especificada no campo Tensão Para Definição de Carga;
- Carga Reativa: Valor da carga reativa da barra, em Mvar.No caso da carga variar com a magnitude da tensão da barra, entre neste campo o valor da carga para a tensão especificada no campo Tensão Para Definição de Carga;
- Capacitor Reator: Valor total da potência reativa injetada na barra, em Mvar, por bancos de capacitores/reatores. O valor a ser preenchido neste campo refere-se a potência reativa injetada na tensão nominal (1.0 p.u.). Este valor deve ser positivo para capacitores e negativo para reatores;
- Área: Número da área à qual pertence a barra;
- Tensão para definição de carga: Entre neste campo com o valor em p.u. da tensão para a qual foi medido o valor das parcelas ativa e reativa da carga definidos nos campos Carga Ativa e Carga Reativa, respectivamente;
- Modo de visiualização: Entre neste campo com o modo de visualização da barra CA no diagrama unifilar: 0 para barra normal, 1 para barra midpointe 2 para barra auxiliar;
- Agregador 1-5: Entrar neste campo com o número da ocorrência do agregador genérico 1-5 à qual a barra CA está associada;
- *: Para as barras de referência, a geração reativa mínima e máxima serão -9999.0 e 99999.0 Mvar, em outros casos será 0 Mvar;
- **: O valor será a própria barra.

O código de execução DLIN apresenta os dados de circuito CA, sendo que cada linha de informa os dados de uma linha ou transformador específico, enquanto as colunas são agrupadas em campos que indicarão esses dados por tipo. Os campos, bem como as

colunas e seus valores padrão são apresentados na Tabela 2.2. É importante detalhar os campos apresentados na Tabela 2.2 (ANAREDE, 2019).

Tabela 2.2 – Descrição dos dados em DLIN.

Campo	Colunas	Default
Da Barra	01-05	
Abertura da Barra	06-06	L
Operação	08-08	A
Abertura Para Barra	10-10	L
Para Barra	11-15	
Circuito	16-17	*
Estado	18-18	L
Proprietário	19-19	F
Resistência	21-26	0.0
Reatância	27-32	
Susceptância	33-38	0.0
Tap	39-43	
Tap Mínima	44-48	
Tap Máxima	49-53	
Defasagem	54-58	0.0
Barra controlada	59-64	**
Capacidade Normal	65-68	$\inf$
Capacidade em Emergência	69-72	Cap. Normal
Número de Taps	73-74	33
Capacidade de Equipamento	75-78	Cap. Normal
Agregador 1-10	79-108	

Fonte: Adaptado de ANAREDE (2019).

- Da Barra: Número da barra de uma das extremidades do circuito como definido no campo Número do Código de Execução DBAR;
- Abertura da Barra: L para ligado e D para desligado;
- Operação: A ou 0 para adição de dados de circuito, E ou 1 para eliminação de dados de circuito e M ou 2 para modificação de dados de circuito;
- Abertura Para Barra: L para ligado e D para desligado;
- Para Barra: Número da barra da outra extremidade do circuito como definido no campo Número do Código de Execução DBAR;

- Circuito: Número de identificação do circuito CA em paralelo;
- Estado: L se o circuito estiver em operação (ligado). D se o circuito estiver fora de operação (desligado);
- Proprietário: F se o circuito pertencer a área da barra definida no campo Da Barra e T se o circuito pertencer a área da barra definida no campo Para Barra. As perdas de potência ativa nos circuitos são contabilizadas para a área a qual pertence o circuito (definido pelo campo proprietário) e, para efeito de intercâmbio, os fluxos são calculados na extremidade conectada à barra da área não proprietária do circuito;
- Resistência: Valor da resistência do circuito, em %. Para transformadores este valor corresponde ao valor da resistência para o tap nominal;
- Reatância: Valor da reatância do circuito, em %. Para transformadores este valor corresponde ao valor da reatância para o tap nominal;
- Susceptância: Valor total da susceptância shunt do circuito, em Mvar;
- Tap: Valor do tap referido à barra definida no campo Da Barra, em p.u., para os transformadores de tap fixo ou, uma estimativa deste valor para os transformadores com variação automática de tap (LTC). Os transformadores tipo LTC são identificados pelo preenchimento dos campos Tap Mínimo e Tap Máximo. Nesse caso, se o valor inicial do tap não for especificado, o valor 1.0 p.u. é considerado. Se o valor inicial do tap estiver fora dos limites especificados, este valor é considerado igual ao valor do limite violado;
- Tap Mínima: Valor mínimo que o tap pode assumir, em p.u., para transformadores com variação automática de tap;
- Tap Máxima: Valor máximo que o tap pode assumir, em p.u., para transformadores com variação automática de tap;
- Defasagem: Valor do ângulo de defasamento, em graus, para transformadores defasadores. O defasamento angular especificado é aplicado em relação ao ângulo da barra definido no campo Da Barra;
- Barra controlada: No caso de circuitos tipo transformador com variação automática de tap, este campo é destinado ao número da barra cuja magnitude da tensão deve ser controlada;
- Capacidade Normal: Capacidade de carregamento do circuito em condições normais para fins de monitoração de fluxo, em MVA;

- Capacidade em Emergência: Capacidade de carregamento do circuito em condições de emergência para fins de monitoração de fluxo, em MVA;
- Número de Taps: Número de posições do transformador de tap variável, incluindo o tap mínimo e o tap máximo;
- Capacidade de Equipamento: Capacidade de carregamento do equipamento com menor capacidade de carregamento conectado ao circuito;
- Agregador 1-10: Entrar neste campo com o número da ocorrência do agregador genérico 1-10 à qual o circuito está associado;
- *:No caso de adição de dado de circuito o valor *default* para o número do circuito em paralelo consiste do primeiro número disponível a partir do maior número do circuito em paralelo cujo dado já existe. No caso de alteração ou eliminação o valor *default* é igual ao menor número do circuito em paralelo;
- **: Se a barra controlada não for uma das barras definidas nos campos Da Barra ou Para Barra, deve ser associado um sinal ao número desta barra que determine a direção do movimento do tap no sentido de aumentar a magnitude da tensão da barra controlada. Em geral, barras situadas no lado do tap (Da Barra), recebem um sinal positivo e, barras situadas no lado contrário do tap (Para Barra), recebem um sinal negativo.

O código de execução de DGBT apresenta os grupos de base de tensão de barras CA, sendo que, quando existe DGBT, cada linha apresenta uma base com sua nomenclatura e se respectivo valor de tensão associada, enquanto as colunas são agrupadas em campos que indicarão esses dados por tipo. Os campos, bem como as colunas e seus valores padrão são apresentados na Tabela 2.3. Os campos apresentados na Tabela 2.3 são detalhados abaixo (ANAREDE, 2019).

Tabela 2.3 – Descrição dos dados em DGBT.

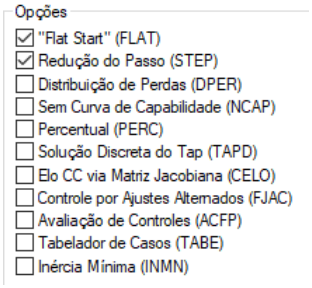
Campo	**Colunas**	**Default**
Grupo	01-02	0
Tensão	04-08	1.0

Fonte: Adaptado de ANAREDE (2019).

- Grupo: Identificador do grupo base de tensão, como definido no campo Grupo Base de Tensão do Código de Execução DBAR.
- Tensão: Tensão base associada ao grupo, em kV.

Para a simulação do fluxo de potência da rede são consideradas algumas opções apresentadas na Figura 2.1. Com a opção *flat start* (FLAT) as magnitudes das tensões das barras é inicializado com o valor de 1 p.u. e os ângulos das barras serão iguais aos ângulo da barra de referência. A opção redução de passo (STEP) limita os valores absolutos da correções do processo iterativo.

Figura 2.1 – Opções assinaladas no fluxo de potência do ANAREDE.



Fonte: Autor.

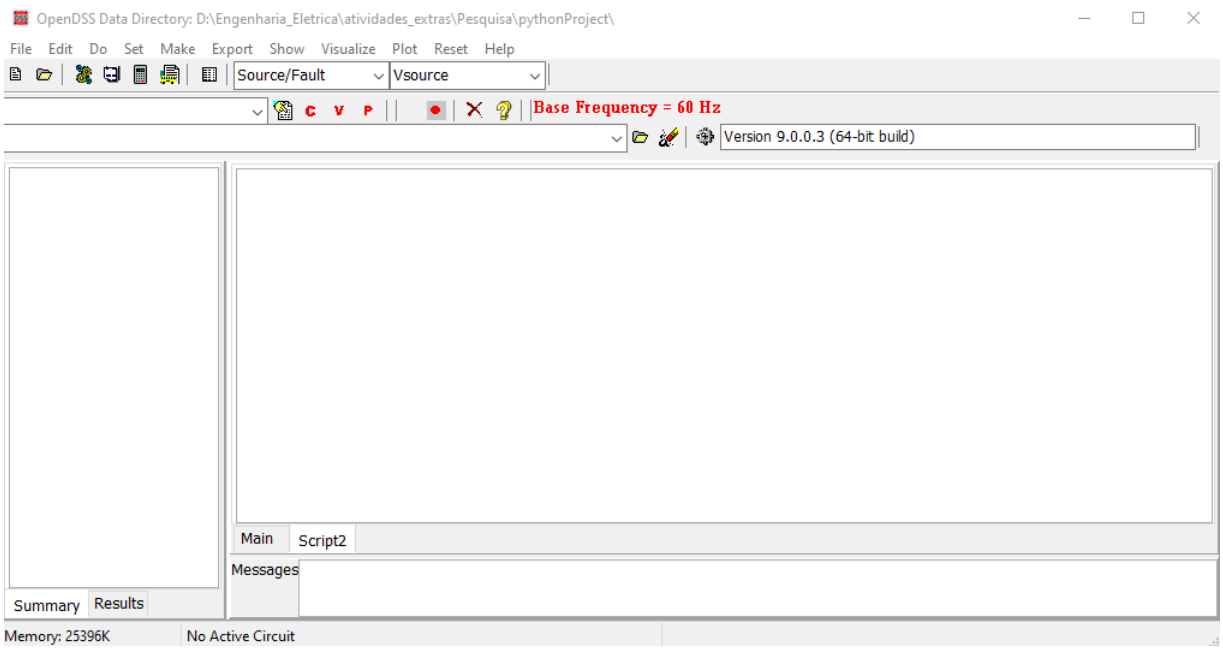
2.3 OPENDSS

O Open Distribuition System Simulator (OpenDSS) começou a ser desenvolvido em Abril 1997 na *Electrotek Concepts, Inc*. O programa se trata de uma ferramenta de simulação para sistemas elétricos de distribuição. Seus usos incluem o planejamento e análise das redes de distribuição, simulações de carga e gerações anuais, estudos probabilísticos, simulações envolvendo painéis fotovoltaicos, análise de interconexões com GD, dentre outros usos. O OpenDSS possui diferentes tipos de soluções para o fluxo de potência tais como *snapshot*, diário e anual, além de soluções dinâmicas, estudos de faltas entre outros tipos de resultados que podem ser obtidos (DUGAN; MONTENEGRO, 2020).

Para utilizar o programa opera-se com um arquivo executável ou Dynamic Link Library (DLL), podendo trabalhar com um servidor Component Object Model (COM) ou não. A versão executável empregada neste trabalho, possui uma interface para usuários baseada em texto. Nela será digitado o *script* da rede e operações desejadas, além de visualizar os resultados (DUGAN; MONTENEGRO, 2020). A tela do programa pode ser visualizada na Figura 2.2.

Anteriormente a 1997 a ênfase era lidar com aspectos distintos do planejamento de sistemas de GD usando métodos de análise convencionais para sistemas de distribuição de energia, sendo que o objetivo era ser uma ferramenta para concessionárias de distribuição, ou mesmo sistemas de subtransmissão e transmissão. Com o tempo o programa foi evoluindo e novas funcionalidades foram adicionadas, ampliando a gama de aplicações.

Figura 2.2 – Tela inicial do programa OpenDSS.



Fonte: Autor.

Atualmente, pode-se solucionar casos radiais ou malhas, com os tamanhos mais variados, sejam pequenos ou médios. Mesmo que análise de GD continue como um uso chave outros usos tem ganho espaço, dentre eles a resolução de problemas de fluxo de potência (DUGAN; MONTENEGRO, 2020).

Os dois tipos básicos de solução de fluxo de potências são com a solução direta ou fluxo de potência iterativo. No primeiro, os elementos não lineares como cargas, geradores distribuídos são incluídos como admitâncias na matriz de admitâncias do sistema. Na forma iterativa, aqueles elementos são tratados como fontes injetadas. Nos caso iterativo, há dois algoritmos que podem ser empregados: modo "Newton" e modo "Normal"de injeção de corrente. O modo Newton não deve ser confundido com o Método de Newton-Raphson, mas se trata de um algoritmo criado para o programa usado em sistemas mais difíceis. O modo "Normal" consiste num método iterativo de ponto-fixo, sendo mais rápido e mais simples (DUGAN; MONTENEGRO, 2020).

Nos sistemas elétricos de distribuição há um série de equipamentos que serão modelados no OpenDSS. Estes podem ser classificados como elementos de entrega de potência, *Power Delivery* (PD), ou ainda em elementos de conversão de potência, *Power Conversion* (PC). Os PDs agrupam equipamentos como linhas, transformadores, capacitores e reatores, ao passo que os PCs reunem elementos como cargas, geradores, sistemas fotovoltaicos e dispositivos de armazenamento(DUGAN; MONTENEGRO, 2020).

Somado a isso, o programa modela as barras (*bus*), que representam os pontos do circuito onde os elementos são conectados. Contudo, é preciso explicitar que no OpenDSS

as barras não classificadas em PV, PQ ou $V\theta$, sendo modeladas de um modo genérico, isto é, apenas os locais do circuito os elementos se conectam. Além disso, para o software, é feita uma distinção conceitual entre barra e nó. Aqui, um nó é um ponto onde se conectam objetos, a exemplo de uma carga ou linha. A barra, por outro lado, contém os nós. Outro ponto a destacar é que o nó 0 será a referência sempre, ou seja, o terra (*ground*), possuindo uma tensão de $0V$ (DUGAN; MONTENEGRO, 2020).

A solução envolverá a formulação com a admitância por nó, segundo a Equação 2.24. Primeiramente, haverá uma matriz primitiva (Y_{prim} para cada elemento do circuito, sendo que essas matrizes nodais de admitância serão utilizadas para construir a matriz principal do sistema. O valor inicial das tensões para o vetor V é obtido pela resolução direta de $I = Y.V$, com geradores e cargas modelados como equivalentes lineares sem injeção de corrente, algo semelhante ao "*flat start*". O ciclo de iterações terá início quando as correntes injetados de todos os PCs são obtidas para o vetor I_{inj}. O ciclo é repetido até a tensão convergir com um valor 0.0001 p.u. (DUGAN; MONTENEGRO, 2020).

$$I_{inj} = Y_{system}.V \tag{2.24}$$

O OpenDSS ainda possibilita realizar a representação gráfica de valores como a tensão, corrente, potência, perdas, capacidade restante, ressonância harmônica, entre outras informações. Para plotar perfis de tensão é preciso adicionar um medidor de energia com o comando "Energymeter". Outro comando que pode ser destacado é "CalcVoltageBases" onde são calculadas as bases de tensão do sistema, realizando um fluxo de potência sem cargas (DUGAN; MONTENEGRO, 2020).

A sintaxe no software é constituída do comando e dos parâmetros. Estes são separados por vírgulas ou espaços. Os parâmetros podem ter valores atribuídos, seja utilizando o nome do parâmetro e o valor, como por exemplo nome=valor, ou apenas o valor, desde que esteja na posição especifica. Para cada novo comando se requer o "new" e o nome do objeto. Além disso, é possível realizar comentários nos textos utilizando "//" no início da linha ou então "!" para comentar no meio da linha (DUGAN; MONTENEGRO, 2020).

Por fim, destaca-se que o OpenDSS possui um vasto conjunto de opções para solucionar os caso. O tipo de solução tem algoritmos, como Normal ou Newton. O período analisado pode ser instantâneo ("snap"), diário ("daily") ou anual ("yearly"). O tipo de solução que se busca pode ser um fluxo de potência, estudo de falta ("Faultstudy"), solução harmônica ("Harmonics"), solução dinâmica ("Dynamics"), Monte-Carlo ("M1","M2","M3","MF"), dentre outras (DUGAN; MONTENEGRO, 2020).

2.4 PYTHON

A linguagem de programação Python tem sua origem na década de 1980 com o programador holandês Guido van Rossum, que nomeou esta linguagem inspirado na série de comédia Monty Python (PERKOVIC, 2016). Esta linguagem é do tipo orientado ao objeto, ou seja, os dados são representados por objetos, que por sua vez, possuem três aspectos, a saber, o identificador, que especifica o nome do objeto, o tipo que define a natureza dos dados, por exemplo inteiro ("int) ou ponto flutuante ("float"), e o conteúdo que expressa o valor armazenado (BANIN, 2018).

A linguagem Python dispõe de um conjunto de características diferenciadas que a tornam atrativa a sua escolha. Dentre as peculiaridades que podem ser destacadas está a sua natureza *opensource*, isto é, pode ser utilizado e distribuída de maneira livre. Somado a isso, o Python possui o código-fonte está na linguagem ANSI C, o que o torna disponível para vários sistemas operacionais, tais como Linux, Windows, macOS, entre outras, além de que as bibliotecas-padrão são executadas de modo semelhante em qualquer uma dessa plataformas (BANIN, 2018). Salienta-se ainda que o Python é considerada uma linguagem de fácil aprendizado, sendo indicada como uma linguagem inicial para adentrar a programação (PERKOVIC, 2016). O Python é capaz de unificar tanto a simplicidade quanto a robusteza, desse modo possibilitando projetos, dos mais variados, sejam triviais ou complexos e enormes, permitindo trabalho com grande volume de dados e interconexão com interfaces. Dentre as áreas nas quais este tem sido usado pode-se citar aplicações nas ciências, ferramentas para engenharia, jogos e lazer, administração, entre outras (BANIN, 2018).

3 METODOLOGIA

Neste capítulo será tratada da metodologia proposta que permitirá a modelagem e a análise de redes de transmissão juntamente com redes de distribuição de maneira integrada utilizando o software OpenDSS. Primeiro, a metodologia terá sua classificação declarada. Em seguida, será apresentado um panorama geral da conversão do ANAREDE para OpenDSS, depois será detalhado cada etapa dessa conversão. Por fim, será proposta uma forma de integrar as redes e realizar simulações no OpenDSS.

3.1 CLASSIFICAÇÃO DA METODOLOGIA

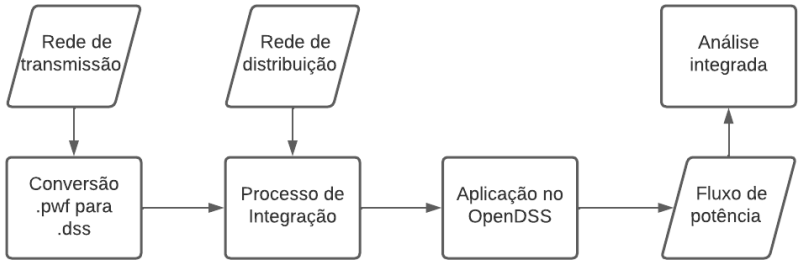
O desenvolvimento experimental é um tipo de trabalho sistemático, que faz de conhecimento provenientes da pesquisa ou experiência e visa produzir novos ou melhorados materiais, equipamentos, comportamentos, políticas, sistemas e serviços (GIL, 2018). Sendo assim, o trabalho elaborado pode ser classificado de tal forma, dado que objetiva solucionar um problema específico, a integração de redes de distribuição e redes de transmissão, desenvolvendo um modelo, em especial uma ferramenta computacional, que permita esta integração.

O propósito da pesquisa exploratória é entender sobre um assunto ainda pouco conhecido (SORDI, 2017). Como está sendo desenvolvida uma ferramenta que não foi feita até então, o trabalho pode ser classificado nesse propósito. O método quantitativo é sequencial e comprobatório (SAMPIERI; COLLADO; LUCIO, 2013). Dado o desejo de propor uma metodologia que permite realizar a integração das redes e que possua resultados adequados, este trabalho pode ser considerado segundo este método.

3.2 PANORAMA DA METODOLOGIA PROPOSTA

A metodologia proposta para integração das redes envolverá uma série de etapas. A primeira é a conversão do cartão da rede de transmissão, o qual se encontra no formato .pwf e será convertido para o formato .dss. Em seguida, será preciso realizar um processo de integração das redes onde serão realizadas alterações nos aquivos. Por fim será aplicado no programa OpenDSS este arquivo integrado, onde será obtido o fluxo de potência, o qual será analisado. Este panorama se encontra na Figura 3.1.

Figura 3.1 – Panorama da metodologia de integração e análise.

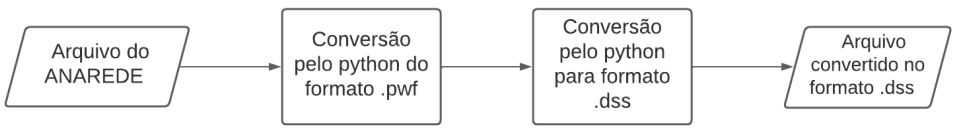


Fonte: Autor.

3.3 CONVERSÃO DO ANAREDE PARA OPENDSS

A implementação da conversão de um cartão do ANAREDE para o OpenDSS será desenvolvida no ambiente Python, onde o cartão no formato .pwf será analisado pelo algoritmo "conversor_pwf_para_py"e depois de ser lido, resultará em algumas matrizes de dados, as quais serão aplicadas em outro algoritmo, "conversor_py_para_dss", que gerará um arquivo no formato .dss. O fluxograma da Figura 3.2 apresenta como a conversão ocorrerá em linhas gerais.

Figura 3.2 – Fluxograma da conversão do arquivo .pwf para o arquivo .dss.



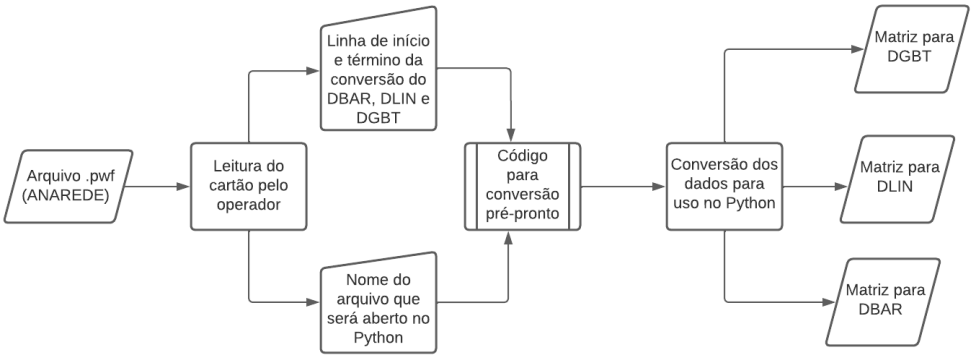
Fonte: Autor.

3.3.1 Conversão do ANAREDE para Python

Primeiramente, o ambiente Python somente trabalhará com os dados do cartão do ANAREDE, após uma conversão inicial. Nesta etapa, os dados se encontram no formato .pwf e serão obtidas três matrizes de dados. A primeira matriz diz respeito as informações de DBAR, a segunda é referente aos dados de DLIN e a terceira, por sua vez, é associada ao DGBT. O fluxograma da Figura 3.3 mostra um panorama desta primeira conversão.

O algoritmo "conversor_pwf_para_py" apresentado no anexo A.1, precisará de al-

Figura 3.3 – Fluxograma da conversão do arquivo .pwf para uso no ambiente Python.



Fonte: Autor.

gumas poucas alterações para ser aplicado corretamente. A primeira informação que precisa ser modificada é o nome do arquivo com sua localização numa pasta, alterando a linha 5 do código. Então, ao inspecionar o cartão do ANAREDE é pode-se extrair outras informações que serão imprescindíveis para a conversão do arquivo, sendo estas as linhas para iniciar e terminar a conversão de DBAR, DLIN e DGBT. Após obter os valor dessas linhas do cartão do ANAREDE, digita-se no algoritmo, nas linhas 33 e 34 para DBAR, 321 e 322 para DLIN e 646 e 647 para DGBT. Caso não haja DGBT colocam-se dois valores iguais para i nas posições (ex:1 e 1),o que implicará na não geração de matriz resultante para DGBT, mesmo que a conversão ainda possa ser executada.

A alteração dos códigos é feita manualmente, pois é uma solução mais simples, levando em conta que as linhas de início e término de conversão podem variar bastante entre os cartões do ANAREDE e que no ambiente Python será reconstruído o cartão .pwf. Cabe destacar que usa-se o valor da linha-1 para a linha inicial, já que o Python começa a contagem de vetores em 0.

Depois de realizar as modificação desses parâmetros no algoritmo, não será preciso proceder com mais alterações no código. O "conversor_pwf_para_py" atuará corretamente, sob a condição de que o cartão não possua informações erradas, ou seja, letras ou números em posições incorretas ou possuindo informações incompletas, a exemplo de faltar o número de termos adequados para a conversão.

O procedimento desenvolvido pelo "conversor_pwf_para_py" será ler o cartão e colocar as informações em um vetor A. Posteriormente, o vetor A é transposto, as linhas se tornam colunas e vice-versa, e o vetor B incorpora estes dados. Em seguida, cria-se uma matriz E que receberá os caracteres de cada linha do cartão separadamente, por exemplo, a palavra "sistema", ficaria como "s", "i", "s", "t", "e", "m", "a". Num terceiro momento, serão

construídas as matriz operando os valores da matriz E. Nesta etapa, cada caractere será combinado conforme a orientação de formato dos dados do ANAREDE. Nesse processo, os caracteres de espaços (’ ’) serão substituídos por ’0’. Por fim, por exemplo, o primeiro termo da matriz ("Kr1") da primeira linha ("Klinha"), recebe os 5 primeiros caracteres que estavam na linha de início da conversão da matriz E.

O DBAR será constituído de 19 elementos por linha, cada um com informações das barras, como o número de identificação, se a barra está ligada ou não, grupo de tensão associada, se há alguma geração ou carga na barra dentre outros dados pertinentes. Cada linha de DBAR estará no formato de strings que posteriormente serão usados pelo outro algoritmo, "conversor_py_para_dss". De maneira semelhante, o DLIN será composto de 23 elementos por linha, cada um representando dados das linhas, como a barra de origem e de destino, se a linha de fato está em operação ou não, os valores de reatância e resistência, a presença de transformadores ou capacitores dentre outros dados relevantes. Por fim, o DGBT será formado por 2 elementos por linha, que indicam a sigla para uma base de tensão e o valor que esta representa. Cada dado estará no formato de "string" e as três matrizes poderão ser empregados no outro código.

É importante ressaltar que, como haverá tanto valores numéricos, quanto palavras que comporão os termos no cartão do ANAREDE, mas optou-se por trabalhar somente com strings e depois no processo de conversão do Python para OpenDSS transformar alguma "string" em número. Além disso, alguns cartões podem não possuir todos os caracteres, o que impediria a conversão, por isso, nesses casos são adicionados valores padrões. Dependendo do tamanho do vetor (t) será reconstruído o termo ou então será definido com um valor padrão.

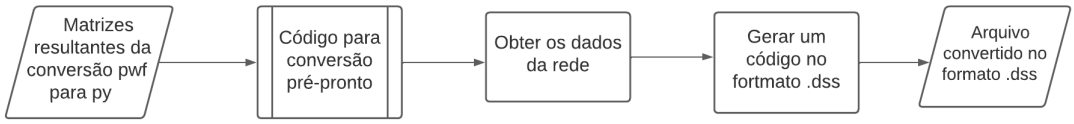
Uma matriz K terá adicionado sobre ela cada uma das linhas ("Klinha"), num processo iterativo até todas as linhas serem geradas. É importante ressaltar que as matrizes das conversões do Python para OpenDSS deverão ter o número de termos iguais por linha, isto explica o motivo de converter as linhas com os dados e colocar numa matriz, eliminando algumas informações que não serão úteis nesta etapa, as quais teriam um tamanho inapropriado para trabalhar. A ideia por traz é que a leitura na matrizes com linhas do mesmo tamanho é o forma como o Python opera, por isso esta preocupação com o número de termos.

Nos anexos é possível visualizar todo o código usado para a conversão do ANAREDE para o ambiente Pyhton. Além disso, estão apresentados os resultados das etapas de conversão do exemplo com o Sistema de 5 barras, como o vetor A, a matriz E, e as matrizes resultantes DBAR e DLIN.

3.3.2 Conversão do Python para OpenDSS

As matrizes provenientes da aplicação do "conversor_pwf_para_py" serão utilizadas na conversão do ambiente Python para o uso no OpenDSS, sendo empregadas no código "conversor_py_para_dss. Nesta próxima etapa será elaborado o arquivo no formato .dss à partir das informações das matrizes, as quais serão lidas e operadas com o algoritmo. O fluxograma da Figura 3.4 mostra como este processo sucederá.

Figura 3.4 – Fluxograma da conversão das matrizes no Pyhton para arquivo no formato .dss.



Fonte: Autor.

O primeiro valor que precisa ser definido é a base de potência. Ele é encontrado em um outro código de execução, o DCTE, porém como daquele conjunto de dados importaria apenas um valor é mais prático apenas alterar a linha 21, com o valor da base, conforme o valor apresentando no cartão. Mesmo assim o código pode adicionar uma base padrão de potência caso não haja essa matriz DCTE, sendo este valor 100 MVA.

Define-se então o "Circuit". É realizada uma varredura em DBAR e a barra que possui a indicação de ser uma V_θ, isto é, que possui um '2' no quarto elemento da linha da matriz DBAR, será utilizada para escrever o DBAR. Obtém-se os dados como nome, o número desta barra, o valor em p.u. da tensão, o ângulo. Então, a base de tensão da barra de referência é obtida verificando, primeiramente se existe a matriz DGBT, a qual recebe os valores das tensões das barras, ou colocando um valor padrão, que o ANAREDE define como 1kV.

Cria-se uma matriz de bases de tensão, este passo ajudará a saber a tensão das barras e demais elementos conectados. Para tanto, verifica-se a existência da matriz DGBT e caso ela exista, ela será relacionada com matriz DBAR, do contrário serão usados valores padrões, definidos pelo ANAREDE como 1kV. Em ambas as situações é criada uma matriz GB, que associará cada barra a um valor de tensão. Ao fim, o algoritmo criará um vetor ordenado e sem repetições com os valores das tensões da base que será usado no comando "calcvoltage bases".

Em seguida, serão obtidos os dados dos transformadores. Verifica-se a existência de transformadores na matriz DLIN, realizando uma varredura para identificar quais elementos possuem tap, indicado pelo décimo segundo elemento da linha da matriz. Além

disso, é confirmado se o transformador está conectado de fato, o que é apontado pelo sétimo elemento da linha da matriz. Sabendo quais linhas de DLIN, representam transformadores, obtém-se a reatância, a resistência, o valor de tap inicial, bem como o número de taps. Também encontra-se a potência do transformador, e nas situações onde não é informado, no ANAREDE significaria uma capacidade infinita, porém aqui será utilizado uma valor padrão igual a base de tensão potência, isto é, 100 MVA.

Após a definição do transformador, busca-se os dados das linhas. Usando a matriz DLIN, verifica-se quais das linhas da matriz são referentes a linhas elétricas, basicamente averiguando quais não possuem tap e depois de identificar quais estão ligadas, obtém-se os dados da origem e do destino das linhas, ou seja, as barras que elas conectam.

Posteriormente encontra-se os dados relativos a cada uma das linhas, que comporão o "Linecode". No ANAREDE, os valores estão em p.u. e não é levada em conta distância das linhas, no entanto o OpenDSS precisa das informações em e das distância. Para solucionar esse problema é criada uma linha que possuirá um tamanho unitário e um valor de resistência e a reatância equivalentes. Calcula-se, então, a base de impedância associada a barra, e os valores de resistência e a reatância em p.u. são convertidos em valores reais em Ω.

Os próximos elementos são as cargas. Inicia-se uma averiguação na matriz DBAR para encontrar alguma carga nas linhas dessa matriz. Isso é feito ao encontrar um valor diferente de zero no décimo quinto ou décimo sexto elementos da linha da matriz. Verifica-se ainda se a carga está ligada de fato. Então, os valores de potência ativa e reativa são extraídos e o fator de potência é calculado. Lembrando que numa situação com potência ativa igual a zero, o valor do fator de potência é simplesmente definido em 1, para evitar uma divisão por zero no algoritmo.

Em seguida, encontram os parâmetros dos geradores. Verifica-se na DBAR as barras do tipo PV, analisando o quarto elemento da matriz. Depois de verificar se a barra está ligada, o gerador é nomeado e obtém-se a potência ativa e/ou reativa gerada. O valor do fator de potência também é calculado e a mesma consideração sobre divisão por zero das cargas aplica-se aqui.

O último elemento que será encontrado são os capacitores. Verifica-se o décimo primeiro elemento da linha da matriz DLIN, o qual indica a existência de susceptância *shunt*. Obtém-se a barra onde o capacitor está conectado, a tensão desta barra e a potência reativa.

Uma vez que os elementos e seus dados foram obtidos será escrito um arquivo que será usado no OpenDSS. O arquivo pode possuir a extensão .dss, permitindo que seja lido diretamente pelo OpenDSS, ou ainda estar no formato .txt de modo que um leitor de texto, possa lê-lo. É digno de nota que o arquivo gerado será criado na pasta onde o código de conversão se localiza.

O documento iniciará pelo escrita do "circuit", fazendo uso das informações advin-

das das matrizes como o nome da barra, a base de potência e a base de tensão e o valor da tensão em módulo, em p.u., e ângulo. O valor de fase é definido em 3 e a frequência em $60Hz$, dado que são os valores usuais de trabalho no Brasil. Dados que não forem explicitados, receberão o valor padrão definido pelo OpenDSS.

No momento da obtenção dos dados de cada um dos equipamentos é preciso averiguar sobre a existência, bem como o número destes. Saber isso é essencial, pois a escrita de cada componente da rede levará isso em conta, ou seja, se não houver, por exemplo, geradores, a escrita dos geradores não ocorrerá, ou ainda, se houver três transformadores, serão escritas as informações dos três equipamentos. Para que e o tipo e número de equipamentos seja gerado corretamento efetuam-se iterações conforme o número de equipamentos do tipo especificado e cada elemento é nomeado conforme uma convenção que considera o número da iteração que o elemento situa-se. Por exemplo, uma carga será denominada de acordo com a convenção Ci, sendo que "C"indica que é uma carga e "i"representa o número da iteração.

O segundo tipo de elemento a ser produzido no arquivo serão os transformadores. Para cada equipamento colocam-se as informações de resistência, reatância e número de taps, estabelecendo que os transformadores serão do tipo de 2 enrolamento, que é o tipo padrão do OpenDSS. No primeiro enrolamento é preciso incluir dados como barra de onde o transformador saiu, o tap, a tensão e a potência. Para o segundo procedesse da mesma maneira, com exceção de que será informada a outra barra onde o transformador será conectado. Cabe ressaltar que o tipo de conexão será estrela, que é a conexão padrão do OpenDSS. Cada transformador será nomeado como TRi.

Para o "linecode" são adicionados os dados da resistência e a reatância do cabo, sendo apenas valores de sequência positiva. A unidade de comprimento é indicada como quilômetros. Mesmo que exista cabos que possuirão valores iguais, optou-se por criar um "linecode"para cada linha, isto é, um tipo de cabo associada a cada linha, sendo esta uma solução mais simples, uma vez que a linha ("line") e o cabo ("linecode") são informações separadas. Cada "linecode"será denominado como $Caboi$.

Gera-se então as informações das linhas, no qual há são assinaladas barras que as linham conectam, no OpenDSS denominado de barra de saída e barrada de chegada. As características dessa linha, estão associada ao cabo que ela utiliza, isto é, o "linecode"associado a ela, com um comprimento da linha de 1 quilômetro. A explicação para criar uma linha com valor unitário de distância é devido a modelagem do ANAREDE em comparação com o OpenDSS. Enquanto o ANAREDE apresenta os valores em p.u. e não indica as distâncias das barras, o OpenDSS usa os valores reais para o "linecode", isto é, Ω/km e a é a impedância da linha é calculada com a distância informada das barras. Para solucionar isso, uma rede equivalente foi criada com distância de 1km, mas com a impedância total da linha em Ω/km. As linhas são designadas como Li.

O próximo elemento a ser escrito são as cargas. Primeiro, esta é nomeada, con-

forme o padrão estipulado, Ci, e então a localização dela é apontada, ou seja, a qual barra a carga está conectada. Depois são informados os valores da tensão da barra, o valor de potência ativa, da potência reativa e do fator de potência. Os demais detalhes sobre o que a carga poderia ter receberão os valores padrão do OpenDSS.

Em seguida, o gerador será modelado de acordo com o tipo "model=1" , isto é, com uma injeção de potência ativa e reativa constante. Este modelo foi selecionado, devido aos valores das simulações obterem um resultado mais próximo dos valores no ANAREDE. Posteriormente, no capítulo de Estudo de Caso, serão realizados testes com o "model=3" , o qual indicaria uma injeção de potência ativa e tensão constante. Além do modelo, é indicada a barra na qual o gerador está conectado, a base da tensão e os valores das potências. Os demais valores estarão nas pré-definições do OpenDSS. Cada gerador é nomeado como Gi.

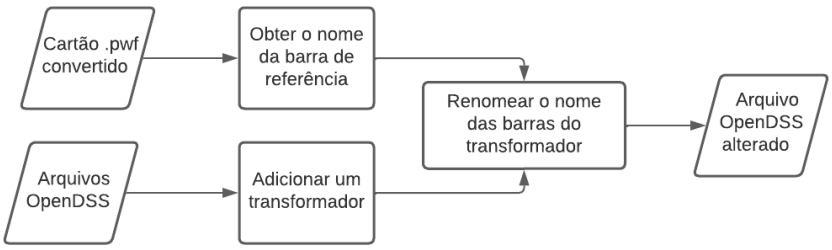
O capacitor é último equipamento a ser escrito no arquivo, sendo que será denominado como $CAPi$, e posteriormente é informado a barra onde o capacitor *shunt* está localizado, depois acrescenta-se a a tensão da barra e potência reativa. O número de fases é designado como 3 e demais valores assumem o padrão do OpenDSS.

Para finalizar, são criadas algumas linhas que especificam o tipo de solução, que na corrente aplicação será o modo "snapshot". Adiciona-se os valores das bases de tensão para o cálculo de bases. Por fim, coloca-se o comando para solucionar ("solve") a rede gerado. Então o arquivo criado é encerrado no algoritmo.

3.4 INTEGRAÇÃO DA REDE DE TRANSMISSÃO E A REDE DE DISTRIBUIÇÃO NO OPENDSS

Uma vez que um arquivo do ANAREDE foi convertido para uso no OpenDSS é possível utilizá-lo em conjunto com demais arquivos do OpenDSS. Assim uma rede de transmissão que estava modelada no ANAREDE, agora foi modelada no OpenDSS. Como a rede de distribuição já está modelada no OpenDSS, basta conectá-las e realizar as simulações. O processo de integração pode ser visualizado na Figura 3.5.

Figura 3.5 – Fluxograma do processo de integração das redes.



Fonte: Autor.

Para realizar a conexão sugere-se que a barra de "circuit" proveniente do ANAREDE, isto é, da rede de transmissão, torne-se a barra de referência da rede distribuição. Dado que a rede de transmissão teria uma tensão mais elevada que a distribuição, será necessário adicionar um transformador que permita essa conexão. Será preciso ajustar nomes das barras para que elas coincidam, assim como digitar os valores das tensões das redes . Estes processos deverão ser realizados manualmente pelo usuário.

4 ESTUDO DE CASOS

4.1 VALIDAÇÃO DA CONVERSÃO DO ANAREDE PARA OPENDSS

Visando constatar se a conversão dos cartões advindos do ANAREDE para o OpenDSS foi realizada de maneira adequada serão realizados testes com duas redes disponibilizados pelo próprio CEPEL na biblioteca de exemplos do ANAREDE. Os exemplos utilizados serão os sistemas de cinco barras e outro de nove barras. A escolha desses exemplos é devido a simplicidade das redes o que pode facilitar as análises nelas. Depois que a conversão for realizada serão comparados os valores de fluxo de potência obtidos nos dois softwares.

4.1.1 Sistema de 5 barras

4.1.1.1 Conversão de .pwf para o ambiente Python

À título de facilitar a visualização do processo de conversão, será realizada a explicação juntamente com a aplicação da conversão com o Sistema de 5 barras presente nos exemplos do ANAREDE. O cartão com os dados desse sistema é apresentado na Figura 4.1, onde é usado o software Notepad++ para visualizar o arquivo .pwf.

Figura 4.1 – Cartão do ANAREDE do Sistema de 5 barras.

SIST5BARRAS.PWF

```
TITU
Caso Sistema Teste 5 Barras
DBAR
(Num)OETGb(   nome   )Gl( V)( A)( Pg)( Qg)( Qn)( Qm)(Bc  )( Pl)( Ql)( Sh)Are(Vf)
    1 L2  Maq1          1017  0.  30.6.522-99999999.    30                  11000
    2 L1  Maq2          10253.66  90.19.56-99999999.    30                  11000
   10 L   Barra_Term_1  1006-3.4                                            11000
   20 L   Barra_Term_2  1014 .18                                            11000
   30 L   Barra_PQ      1000-9.5                           120.   0.        11000
99999
DLIN
(De )d O d(Pa )NcEP  ( R% )( X% )(Mvar)(Tap)(Tmn)(Tmx)(Phs)(Bc  )(Cn)(Ce)Ns
    1        10 1       .1   20.
    2        20 1       .1    7.
   10        20 1       .1   14.       1.02
   10        20 2       .1   14.         .9
   10        30 1       .1   18.
   10        30 2       .1   18.
99999
FIM
```

Fonte: Autor.

O algoritmo utilizado nessa etapa é denominado "conversor_pwf_para_py". Este já está pré-construído, necessitando apenas alterar alguns dados das linhas para realizar a

conversão corretamente. A primeira informação que precisa ser adicionada é o nome do arquivo, sendo assim, altera-se a linha 5 do código conforme a Figura 4.2.

Figura 4.2 – Primeira etapa do "conversor_pwf_para_py".

```
1  import numpy as np
2  import math
3
4  #Definir qual arquivo será convertido
5  nome = list(open("SIST5BARRAS.PWF","r"))      # Abrir na pasta o arquivo que será convertido
6  nome = [s.rstrip() for s in nome]             # Remove \n no fim de cada linha
7  A = np.array(nome)                            # Criar um vetor com o arquivo
8  #print(A)
9  n=len(A)                                      # Obtem o número de itens em A
10 #print(n)
11 B=A.T                                         # Realiza a transposição de A
12 #print(B)
```

Fonte: Autor.

Realizando uma inspeção visual no cartão do ANAREDE é possível obter alguns dados que serão necessários para a conversão do arquivo, sendo estes as linhas para iniciar e terminar a conversão de DBAR e conversão de DLIN, de acordo com a Figura 4.3. A linha de início da conversão DBAR é a seta vermelha, a linha de término da conversão do DBAR é a seta azul. A linha de início da conversão DLIN é a seta verde, a linha de término da conversão do DLIN é a seta roxa. Como não há DGBT, não haverá matriz resultante para DGBT, muito embora a conversão ainda poderá ocorrer.

Figura 4.3 – Linhas para conversão do DBAR e DLIN.

SIST5BARRAS.PWF

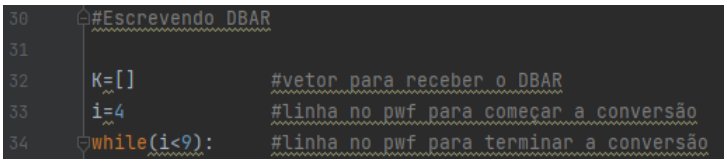
```
1  TITU
2  Caso Sistema Teste 5 Barras
3  DBAR
4  (Num)OETGb(    nome    )Gl( V)( A)( Pg)( Qg)( Qn)( Qm)(Bc   )( Pl)( Ql)( Sh)Are(Vf)
5      1 L2  Maq1          1017  0.  30.6.522-99999999.    30                  11000
6      2 L1  Maq2          10253.66  90.19.56-99999999.    30                  11000
7     10 L   Barra_Term_1  1006-3.4                                            11000
8     20 L   Barra_Term_2  1014 .18                                            11000
9     30 L   Barra_PQ      1000-9.5                          120.   0.         11000
10 99999
11 DLIN
12 (De )d O d(Pa )NcEP  ( R% )( X% )(Mvar)(Tap)(Tmn)(Tmx)(Phs)(Bc   )(Cn)(Ce)Ns
13     1         10 1        .1   20.
14     2         20 1        .1    7.
15    10         20 1        .1   14.        1.02
16    10         20 2        .1   14.          .9
17    10         30 1        .1   18.
18    10         30 2        .1   18.
19 99999
20 FIM
21
```

Fonte: Autor.

Após a inspeção visual do cartão do ANAREDE, alteram-se os valores de início e término da conversão de DBAR e DLIN no código "conversor_pwf_para_py", enquanto as linhas de DGBT serão iguais a zero. Esta alteração é indicada nas Figuras 4.4, 4.5 e 4.6. Este processo é feito manualmente devido a praticidade, uma vez que as linhas de início e término de conversão podem variar consideravelmente entre os cartões do ANAREDE.

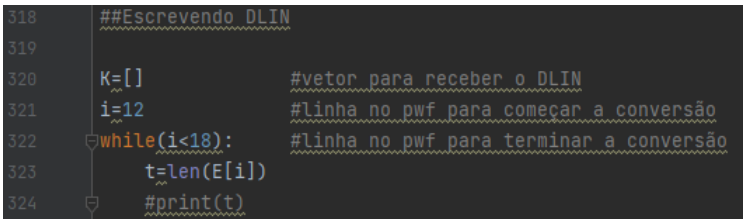
É válido ressaltar que usa-se o valor da linha igual 1 para a linha inicial, uma vez que o Python começa a contagem de vetores em 0, depois 1 e assim por diante.

Figura 4.4 – Linhas para conversão do DBAR.



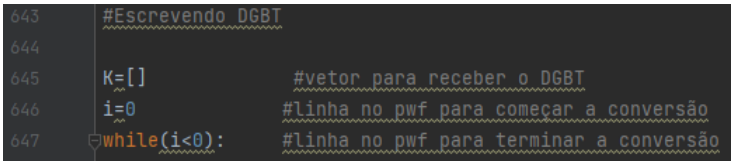
Fonte: Autor.

Figura 4.5 – Linhas para conversão do DLIN.



Fonte: Autor.

Figura 4.6 – Linhas para conversão do DGBT.



Fonte: Autor.

Depois de colocar manualmente esses parâmetros, não há mais necessidade de realizar alterações no código. O "conversor_pwf_para_py" deverá atuar corretamente, contanto que o cartão não possua informações erradas, isto é, letras ou números em posições que não deveriam estar ou com informações incompletas, como faltar o número de termos adequados para a conversão.

O processo que o "conversor_pwf_para_py" realizará será basicamente ler o cartão e colocar as informações em um vetor A. Em seguida, é realizada a transposição desse vetor A e o vetor B receberá esses valores. Na segunda etapa, então, será criada uma matriz E, que receberá cada caractere separado que havia em cada linha do cartão, o que pode ser visualizada na Figura 4.7.

Figura 4.7 – Segunda etapa do "conversor_pwf_para_py".

```
#Separar cada termo
i=0
C=[]
E=[]                    # Matriz que receberá todos caracteres do arquivo
while(i<n):
    C = B[i]
    m=len(C)
    j=0
    linhaE = []
    while (j<m):
        linhaE.append(C[j])
        j = j + 1
    E.append(linhaE)
    i = i + 1
#print(E)
```

Fonte: Autor.

Na terceira etapa, serão construídas duas matriz utilizando os valores da matriz E. Nesse momento, cada caractere será reunido conforme o formato dos dados que o ANAREDE orienta. Assim, por exemplo, o primeiro termo da matriz ("Kr1") da primeira linha ("Klinha"), recebe os 5 primeiros caracteres que estavam na linha de início da conversão da matriz E, conforme a Figura 4.8. Nesse processo, os caracteres de espaços (' ') serão substituídos por '0'.

Figura 4.8 – O primeiro termo de uma linha para DBAR.

```
# Número
K1 = []
N = [E[i][0], E[i][1], E[i][2], E[i][3], E[i][4]]   # Número de identificação da barra CA.
#print(N)
m = len(N)
j = 0
while (j < m):
    if N[j] != ' ':
        K1.append(N[j])
    j = j + 1
    Kr1= ''.join(K1)
#print(Kr1)
```

Fonte: Autor.

Nessa etapa serão gerados todos os termos de cada uma das linhas de DBAR e DLIN. É importante ressaltar que, como haverá tanto valores numéricos, quanto palavras que comporão os termos, optou-se por trabalhar somente com strings e depois no processo de conversão do Python para OpenDSS transformar alguma "string" em número. Além disso, alguns cartões podem não possuir todos os caracteres, o que impediria a conversão, por isso, nesses casos são adicionados valores padrões. Dependendo do tamanho do vetor (t) será reconstruído o termo ou então será definido com um valor padrão.

Após obter cada um dos termos de uma linha, uma matriz K receberá está linha e o processo recomeçará até que todas as linhas de dados sejam escritas, como na Figura

Figura 4.9 – Definindo um valor padrão ou reconstruindo o termo.

```
274     Kr18 = '0'          # Coloca um valor padrão
275     Kr19 = '0'          # Coloca um valor padrão
276
277     if t > 73:
278         # Área
279         K18 = []
280         Area = [E[i][73], E[i][74], E[i][75]]
281         # print(Area)
282         m = len(Area)
283         j = 0
284         while (j < m):
285             if Area[j] != ' ':
286                 K18.append(Area[j])
287             j = j + 1
288             Kr18 = ''.join(K18)
289         if len(K18) == 0:
290             Kr18 = '0.'
291         # print(Kr18)
```

Fonte: Autor.

4.10 . Cabe destacar que as conversões do Python para OpenDSS envolverão matrizes quadradas. Está é uma das razões, pelas quais serão convertidos apenas as linhas com os dados de DBAR, e posteriormente de DLIN, já que elas terão o mesmo número de caracteres, ao invés das linhas com textos que não possuem relevância para esta conversão.

Figura 4.10 – Criando a matriz DBAR.

```
309     Klinha = [Kr1, K2[0], K3[0], K4[0], Kr5, Kr6, Kr7, Kr8, Kr9, Kr10, Kr11, Kr12, Kr13, Kr14,
310               Kr15, Kr16, Kr17, Kr18, Kr19]
311     #print(Klinha)
312
313     K.append(Klinha)
314     i=i+1
315
316 DBAR=K
```

Fonte: Autor.

Nos anexos é possível visualizar todo o código usado para a conversão do ANAREDE para o ambiente Pyhton. Além disso, estão apresentados os resultados das etapas de conversão do exemplo com o Sistema de 5 barras, como o vetor A, a matriz E, e as matrizes resultantes DBAR e DLIN.

4.1.1.2 Conversão do ambiente Python para .dss

As matrizes resultantes, DBAR e DLIN, serão inseridas no código "conversor_py_para_dss" manualmente. Nas Figuras 4.11 e 4.12 estão apresentados as matrizes no código. Como não há informações de DGBT para esse cartão, no código será colocado uma matriz padrão que associará ao valor *default* do ANAREDE, no caso 1 kV. Outro valor que será

colocado como padrão do ANAREDE é a base de potência, a saber 100 MVA, muito embora ela poderia ser alterada, sendo um dado presente no DCTE.

Figura 4.11 – Dados da matriz DBAR do Sistema de 5 barras.

```
# INSERINDO DBAR

DBAR = np.array([["1","0","L","2","0","Maq1","0","1017","0.","30.","6.522","-9999","9999.","30","0","0","0","1","1000"],
["2","0","L","1","0","Maq2","0","1025","3.66","90.","19.56","-9999","9999.","30","0","0","0","1","1000"],
["10","0","L","0","0","Barra_Term_1","0","1006","-3.4","0","0","0","0","0","0","0","0","1","1000"],
["20","0","L","0","0","Barra_Term_2","0","1014",".18","0","0","0","0","0","0","0","0","1","1000"],
["30","0","L","0","0","Barra_PQ","0","1000","-9.5","0","0","0","0","0","120.","0","0","1","1000"]])
```

Fonte: Autor.

Figura 4.12 – Dados da matriz DLIN do Sistema de 5 barras.

```
# INSERINDO DLIN

DLIN = np.array([["1","0","0","0","10","1","0","0",".1","20.","0","0","0","0","0","0","0","0","0","0"],
["2","0","0","0","20","1","0","0",".1","7.","0","0","0","0","0","0","0","0","0","0"],
["10","0","0","0","20","1","0","0",".1","14.","0","1.02","0","0","0","0","0","0","0","0"],
["10","0","0","0","20","2","0","0",".1","14.","0",".9","0","0","0","0","0","0","0","0"],
["10","0","0","0","30","1","0","0",".1","18.","0","0","0","0","0","0","0","0","0","0"],
["10","0","0","0","30","2","0","0",".1","18.","0","0","0","0","0","0","0","0","0","0"]])
```

Fonte: Autor.

O código então obterá os dados do sistema elétrico à partir das informações nas matrizes. Será verificado a presença de diversos equipamentos como transformadores, geradores, cargas, banco de capacitores bem como suas características. As linhas que conectam cada barra serão identificadas e seus valores reais de impedância serão calculados. Por fim, um documento será criado na própria pasta onde o código se encontra com todas as informações relevantes. O código completo pode ser visualizado na sessão de anexos. O documento resultante pode ser visualizado na Figura 4.13.

4.1.1.3 Resultados

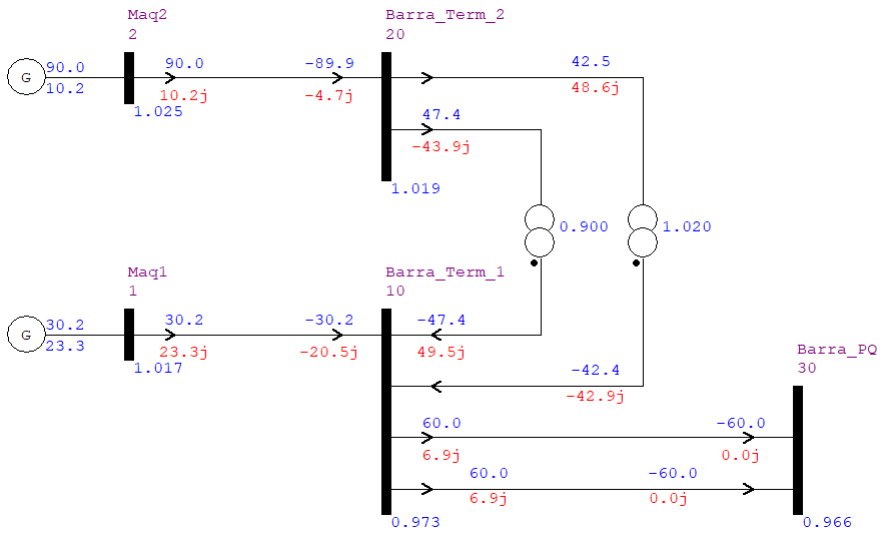
Para verificar a validade da conversão comparam-se os valores resultantes da simulação no ANAREDE e no OpenDSS. Na Figura 4.14 está apresentada a rede após rodar o fluxo de potência com valores de potência ativa e reativa nos elementos. Destacam-se os valores no gerador da barra da referência de 30,2 MW e 23,3 Mvar. O resultado do fluxo de potência simulado no OpenDSS pode ser visualizado na Figura 4.15. Ressalta-se o valor de 30,48 MW e 12,82 Mvar na barra de referência. Na seção de discussão dos resultados será explicada a discrepância de valores na potência reativa e uma forma de solucionar. Na seção B dos anexos é possível verificar as tensões dos equipamentos, bem como as potências que percorrem no sistema.

Figura 4.13 – Cartão da rede de 5 barras convertida para OpenDSS.

```
Clear

// ------------------------------- Novo Circuito ----------------------------

New circuit.Maq1 bus1=1 basekv=1 angle=0.0 pu=1.017 phases=3 frequency=60

// ------------------------------- Transformador ----------------------------

New transformer.TR0 XHL=14. %r=0.05 numtaps=0 windings=2
~ wdg=1 bus=10 tap=1.02 conn=wye kv=1 kva=100000
~ wdg=2 bus=20 conn=wye kv=1 kva=100000

New transformer.TR1 XHL=14. %r=0.05 numtaps=0 windings=2
~ wdg=1 bus=10 tap=.9 conn=wye kv=1 kva=100000
~ wdg=2 bus=20 conn=wye kv=1 kva=100000

// ------------------------- Linecode -------------------------------

New Linecode.Cabo0 R1=1e-05 X1=0.002 Units=km

New Linecode.Cabo1 R1=1e-05 X1=0.0007 Units=km

New Linecode.Cabo4 R1=1e-05 X1=0.0018 Units=km

New Linecode.Cabo5 R1=1e-05 X1=0.0018 Units=km

// ------------------------------ Linhas ------------------------------------

New Line.L0 bus1=1 bus2=10 linecode=Cabo0 length=1 units=km

New Line.L1 bus1=2 bus2=20 linecode=Cabo1 length=1 units=km

New Line.L2 bus1=10 bus2=30 linecode=Cabo4 length=1 units=km

New Line.L3 bus1=10 bus2=30 linecode=Cabo5 length=1 units=km

// ------------------------------ Cargas ------------------------------------

New load.C4 bus1=30 kv=1 kw=1.2e+05 kvar=0.0 pf=1 conn=wye model=1 phases=3

// ------------------------------ Gerador -----------------------------------

New generator.G1 bus1=2 kv=1 kw=90000.0 kvar=19560.0 model=1 phases=3

// ---------------------------------- Tipo de Solução ---------------------

Set mode=snapshot

// ------------------------------- Valores bases de tensão ----------------

Set voltagebases=[ 1]
calcvoltagebases

// ---------------------------------- Solucionar --------------------------

Solve
```

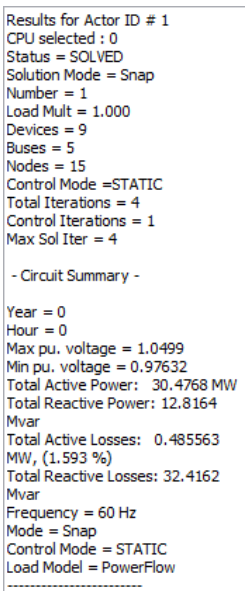
Fonte: Autor.

Figura 4.14 – Fluxo de potência da redede 5 barras no ANAREDE.



Fonte: Autor.

Figura 4.15 – Resultado da simulação da rede de 5 barras no OpenDSS.



Fonte: Autor.

4.1.1.4 Discussão dos resultados

Na simulação do caso do Sistema de 5 barras no OpenDSS encontra-se o valor de 30,48 MW e 12,82 Mvar na barra de referência. O valor de potência ativa é muito semelhante ao da barra de referência simulada no ANAREDE, a saber 30,2 MW, contudo o valor da potência reativa foi consideravelmente diferente. Para solucionar esta discrepância há duas formas.

Na primeira maneira atualiza-se o valor da potência reativa do gerador com o valor do ANAREDE, sendo 10,2 Mvar, mantendo o tipo de modelagem do gerador, neste caso "model=1", com potência ativa e reativa constantes. O resultado do fluxo de potência se encontra na Figura 4.16. Pode-se constatar que o valor das potências ativa e reativa na barra de referência são 30,01 MW e 23,90 Mvar, respectivamente. Com isso é possível inferir que a conversão dos elementos ocorre de maneira adequada, desde que os valores estejam bem ajustados no cartão.

Figura 4.16 – Resultado da simulação da rede de 5 barras com a primeira forma de correção no OpenDSS.

```
Results for Actor ID # 1
CPU selected : 0
Status = SOLVED
Solution Mode = Snap
Number = 1
Load Mult = 1.000
Devices = 9
Buses = 5
Nodes = 15
Control Mode =STATIC
Total Iterations = 4
Control Iterations = 1
Max Sol Iter = 4

 - Circuit Summary -

Year = 0
Hour = 0
Max pu. voltage = 1.0083
Min pu. voltage = 0.94818
Total Active Power:  30.0097 MW
Total Reactive Power: 23.9014
Mvar
Total Active Losses:  0.495273
MW, (1.65 %)
Total Reactive Losses: 34.098
Mvar
Frequency = 60 Hz
Mode = Snap
Control Mode = STATIC
Load Model = PowerFlow
------------------------
```

Fonte: Autor.

Na segunda maneira de solucionar, altera-se o tipo de modelagem do gerador, nesta situação "model=3", com potência ativa e tensão em p.u. constantes. Vale destacar que para esse método funcionar será preciso aumentar o número de iterações, sendo preciso adicionar uma linha de código para aumentar o número máximo de iterações possíveis. A linha do gerador do arquivo, bem como a linha modificada podem ser visualizadas nas Figuras 4.17 e 4.18, respectivamente. Vale destacar que essas informações não precisam ser alteradas manualmente, uma vez que o código "conversor_py_para_dss" já poderia obter de maneira automatizada.

Figura 4.17 – Linha do gerador no arquivo convertido para OpenDSS.

```
// ---------- Gerador ----------
New generator.G1 bus1=2 kv=1 kw=90000.0 kvar=19600.0 model=1 phases=3
```

Fonte: Autor.

Figura 4.18 – Linha do gerador modificada no arquivo convertido para OpenDSS.

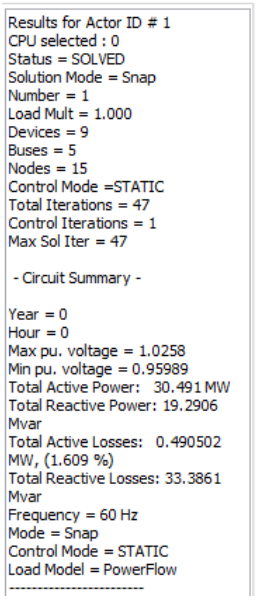
```
// ---------- Gerador ----------
New generator.G1 bus1=2 kv=1 Vpu=1.025 kw=90000.0 model=3 phases=3
```

Fonte: Autor.

O resultado do fluxo de potência se encontra na Figura 4.19. Destaca-se que a potência ativa encontrada foi 30,49 MW, enquanto a potência reativa foi de 19,29 Mvar. O valor da potência reativa foi melhor do que no arquivo convertido inicial, mas não foi um

valor tão próximo quanto o dado obtido na primeira maneira. Além disso, é possível saber o valor da potência reativa do gerador em kVA no OpenDSS, o que está apresentado na Figura 4.20, a saber 14,1 Mvar, sendo um valor diferente dos 10,2 Mvar do gerador no ANAREDE.

Figura 4.19 – Resultado da simulação da rede de 5 barras com a segunda forma de correção no OpenDSS.



Fonte: Autor.

Figura 4.20 – Potência do gerador na segunda forma de correção no OpenDSS do caso com 5 barras.

```
ELEMENT = "Generator.G1"
2        1    -29997.1 +j -4700.7    30363.1       0.9879
2        2    -29997.1 +j -4700.7    30363.1       0.9879
2        3    -29997.1 +j -4700.7    30363.1       0.9879
2        0       0.0 +j    0.0        0.0       1.0000
  TERMINAL TOTAL -89991.2 +j -14102.2    91089.4       0.9879
```

Fonte: Autor.

Os valores obtidos nas diferentes conversões do OpenDSS estão apresentados na Tabela 4.1. O caso com maior semelhança entre a simulação no ANAREDE é o caso onde se corrige as potências do outro gerador, mantendo como modelo de potências ativa e reativa constantes.

Tabela 4.1 – Resultados das simulações do Sistema de 5 barras.

Caso	Pot. Ativa (MW)	Pot. Reativa (Mvar)
ANAREDE	30,2	23,3
OpenDSS	30,48	12,82
OpenDSS 1^a correção	30,01	23,90
OpenDSS 2^a correção	30,49	19,29

Fonte: Autor.

Algumas razões são sugeridas para as diferenças nos valores, sendo a primeira a diferença no algoritmo de cálculo aplicado, uma vez que no ANAREDE é aplicado o método de Newton-Raphson, enquanto no OpenDSS está se aplicando um método iterativo de ponto fixo. Cabe ressaltar que mesmo aplicando o algoritmo de Newton do OpenDSS, que não deve ser confundido com o método de Newton-Raphson, os resultados são muitos semelhantes ao método iterativo de ponto fixo.

Outra razão sugerida é devido ao controle de tensão, uma vez que no ANAREDE é especificada a barra que terá a magnitude da sua tensão controlada com relação a barra informada, enquanto no OpenDSS isto não ocorre. Por exemplo, na barra 2, a barra PV, é indicado que a barra 30, uma barra PQ, terá sua magnitude da tensão controlada. Essa informação nem pode ser expressa no OpenDSS, isto é, dizer que o gerador irá trabalhar com uma potência reativa para atingir um valor de tensão numa barra controlada, dentro dos limites de tensão em p.u. do equipamento.

Com este primeiro exemplo já é possível constatar que a conversão ocorre de maneira satisfatória, atingindo valores semelhantes para a barra de referência. Desde que os dados do cartão em .pwf estejam corretos a conversão e os resultados serão adequados. Inclusive modelando o gerador com potências ativa e reativa constantes implica em valores mais precisos.

4.1.2 Sistema de 9 barras

4.1.2.1 Conversão do cartão do ANAREDE para uso no OpenDSS

Outro exemplo usado para validar a conversão foi o Sistema de 9 barras. Uma parte do cartão em .pwf está apresentada na Figura 4.21. O cartão apresenta outros códigos de execução sendo eles DCTE, DARE, DCTG, DGER e DVSA. O DARE apresenta o intercâmbio de potência ativa entre as áreas, o DCTG lista os casos de contingências, o DGLT mostra os grupos de limites de tensão, o DGER indica os limites de potência ativa e fatores de participação de barras de geração e o DVSA informa grupos de geração que

formam uma região de segurança estática. Esses códigos de execução adicionais não serão convertidos, portanto não serão considerados na simulação do OpenDSS. O vetor A e as matrizes E, DBAR e DLIN estão explicitadas na seção B dos anexos.

Figura 4.21 – Cartão do ANAREDE do Sistema de 9 barras.

```
17 DBAR
18 (Num)OETGb(    nome    )Gl( V)( A)( Pg)( Qg)( Qn)( Qm)(Bc    )( Pl)( Ql)( Sh)Are(Vf)
19     1 L2  Barra 1       11075  0.142.510.88-130.130.4                          11000
20     2 L1  Barra 2       11075-1.8  90.-2.59-101.101.2   7                     21000
21     3 L1  Barra 3       11075-1.4  85.-13.7-67.4 67.4   9                     31000
22     4 L0  Barra 4       11072-4.1                                             41000
23     5 L0  Barra 5       11050-7.7                                125.  50.      41000
24     6 L0  Barra 6       11065-6.7                                 90.  30.      41000
25     7 L0  Barra 7       11078-4.6                                              41000
26     8 L0  Barra 8       11069-6.3                                100.  35.      41000
27     9 L0  Barra 9       11083-3.9                                              41000
28 (   10 L2  Barra 10      11075  0.0.0000.000-999999999                          51000
29 99999
30 DLIN
31 (De )d O d(Pa )NcEP  ( R% )( X% )(Mvar)(Tap)(Tmn)(Tmx)(Phs)(Bc    )(Cn)(Ce)Ns
32     1          4 1 T     0.  5.76        1.0                       250 250
33 (   10          1 1 T     0.  0.01                                   999 999
34     2          7 1 T     0.  6.25        1.0                       200 200
35     3          9 1 T     0.  5.86        1.0                       300 300
36     4          5 1 T     1.   8.5  17.6                            300 300
37     4          6 1 T    1.7   9.2  15.8                            200 200
38     6          9 1 T    3.9   17.  35.8                            200 200
39     7          5 1 T    3.2  16.1  30.6                            300 300
40     7          8 1 T    .85   7.2  14.9                            300 300
41     8          9 1 T   1.19 10.08  20.9                            300 300
42 99999
```

Fonte: Autor.

De maneira análoga ao caso do Sistema de 5 barras primeiramente é preciso encontrar as linhas de início e término da conversão do DBAR e do DLIN. Para DBAR será 18 e 28, enquanto para DLIN será 31 e 41. Aplica-se o código "conversor_pwf_para_py" para gerar as matrizes de DBAR e DLIN e então elas são inseridas no código "conversor_py_para_dss" para gerar o arquivo usado no OpenDSS. O resultado da conversão pode ser visualizado na Figura 4.22.

4.1.2.2 Resultados

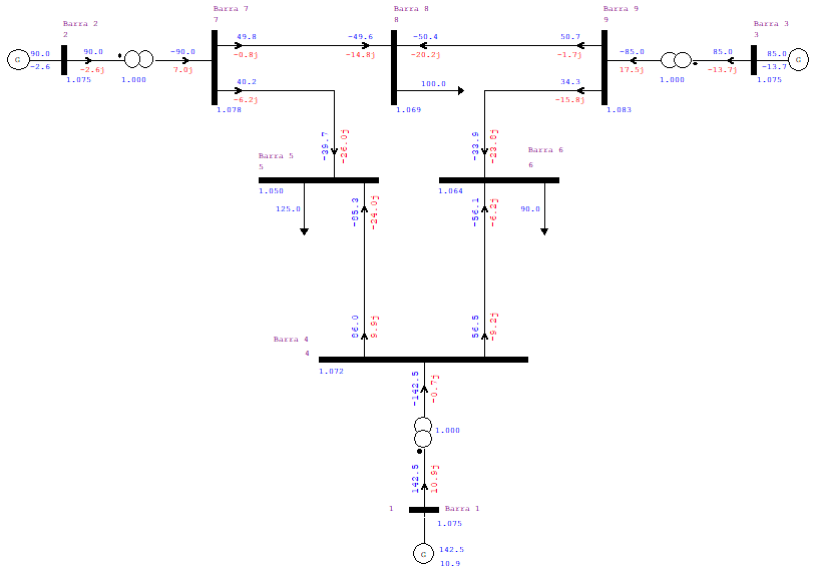
Assim como no caso do Sistema de 5 barras será comparado o resultado do fluxo de potência com o programa ANAREDE e o com o cartão convertido no OpenDSS. Na Figura 4.23 está indicada a rede no ANAREDE, destacando-se aqui os valores de potência ativa e reativa do gerador da barra de referência com 142,5 MW e 10,9 Mvar. No OpenDSS o fluxo de potência resultante, focando no *circuit* pode ser visto na Figura 4.24 o qual mostra 143,44 MW e 10,13 Mvar para a potência da barra de referência. Na seção B dos anexos pode-se verificar as tensões dos equipamentos, bem como as potências que perpassam a rede.

Figura 4.22 – Cartão da rede de 9 barras convertida para OpenDSS.

```
Clear

// ------------------------------ Novo Circuito ------------------------------

New circuit.Barra 1 bus1=1 basekv=1 angle=0.0 pu=1.075 phases=3 frequency=60

// ------------------------------ Transformador ------------------------------

New transformer.TR0 XHL=5.76 %r=0.0 numtaps=0 windings=2
~ wdg=1 bus=1 tap=1.0 conn=wye kv=1 kva=2.5e+05
~ wdg=2 bus=4 conn=wye kv=1 kva=2.5e+05

New transformer.TR1 XHL=6.25 %r=0.0 numtaps=0 windings=2
~ wdg=1 bus=2 tap=1.0 conn=wye kv=1 kva=2e+05
~ wdg=2 bus=7 conn=wye kv=1 kva=2e+05

New transformer.TR2 XHL=5.86 %r=0.0 numtaps=0 windings=2
~ wdg=1 bus=3 tap=1.0 conn=wye kv=1 kva=3e+05
~ wdg=2 bus=9 conn=wye kv=1 kva=3e+05

// ------------------------------ Linecode ------------------------------

New Linecode.Cabo4 R1=0.0001 X1=0.00085 Units=km

New Linecode.Cabo5 R1=0.00017 X1=0.00092 Units=km

New Linecode.Cabo6 R1=0.00039 X1=0.0017 Units=km

New Linecode.Cabo7 R1=0.00032 X1=0.00161 Units=km

New Linecode.Cabo8 R1=8.5e-05 X1=0.00072 Units=km

New Linecode.Cabo9 R1=0.000119 X1=0.001008 Units=km

// ------------------------------ Linhas ------------------------------

New Line.L0 bus1=4 bus2=5 linecode=Cabo4 length=1 units=km

New Line.L1 bus1=4 bus2=6 linecode=Cabo5 length=1 units=km

New Line.L2 bus1=6 bus2=9 linecode=Cabo6 length=1 units=km

New Line.L3 bus1=7 bus2=5 linecode=Cabo7 length=1 units=km

New Line.L4 bus1=7 bus2=8 linecode=Cabo8 length=1 units=km

New Line.L5 bus1=8 bus2=9 linecode=Cabo9 length=1 units=km

// ------------------------------ Cargas ------------------------------

New load.C4 bus1=5 kv=1 kw=1.25e+05 kvar=50000.0 pf=0.928477 conn=wye model=1 phases=3

New load.C5 bus1=6 kv=1 kw=90000.0 kvar=30000.0 pf=0.948683 conn=wye model=1 phases=3

New load.C7 bus1=8 kv=1 kw=1e+05 kvar=35000.0 pf=0.943858 conn=wye model=1 phases=3

// ------------------------------ Gerador ------------------------------

New generator.G1 bus1=2 kv=1 kw=90000.0 kvar=-2590.0 model=1 phases=3

New generator.G2 bus1=3 kv=1 kw=85000.0 kvar=-13700.0 model=1 phases=3

// ------------------------------ Capacitor ------------------------------

New capacitor.CAP0 bus1=4 kv=1 kvar=17600.0 phases=3

New capacitor.CAP1 bus1=4 kv=1 kvar=15800.0 phases=3

New capacitor.CAP2 bus1=6 kv=1 kvar=35800.0 phases=3

New capacitor.CAP3 bus1=7 kv=1 kvar=30600.0 phases=3

New capacitor.CAP4 bus1=7 kv=1 kvar=14900.0 phases=3

New capacitor.CAP5 bus1=8 kv=1 kvar=20900.0 phases=3

// ------------------------------------ Tipo de Solução ------------------------

Set mode=snapshot

// ------------------------------ Valores bases de tensão ------------------------

Set voltagebases=[ 1]
calcvoltagebases

// ------------------------------------ Solucionar ------------------------

Solve
```

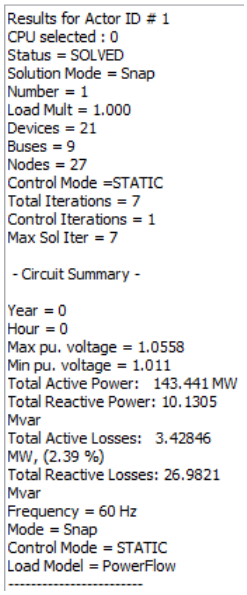
Fonte: Autor.

Figura 4.23 – Fluxo de potência da rede de 9 barras no ANAREDE.



Fonte: Autor.

Figura 4.24 – Resultado da simulação da rede de 9 barras no OpenDSS.



Fonte: Autor.

4.1.2.3 Discussão dos resultados

A conversão do cartão do ANAREDE do sistema de 9 barras teve resultados muitos semelhantes na simulação no OpenDSS quando comparado ao fluxo de potência do ANAREDE. Mesmo assim foram realizadas correções, de forma análogo ao sistema de 5 barras. Constatou-se que a primeira forma de corrigir é desnecessária, uma vez que os valores de potência ativa e reativa dos geradores, após o fluxo de potência, não possuem diferenças significativas com relação aos valores do cartão. Os valores dos casos do Sistema de 9 barras se encontram na Tabela 4.2.

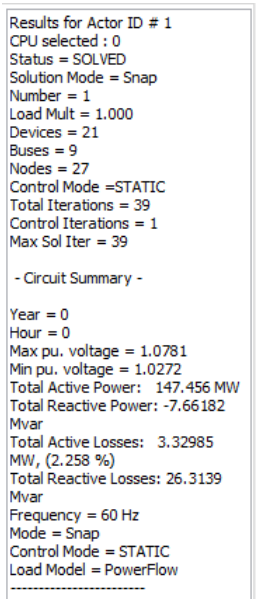
Tabela 4.2 – Resultados das simulações do Sistema de 9 barras.

Caso	Pot. Ativa (MW)	Pot. Reativa (Mvar)
ANAREDE	142,5	10,9
OpenDSS	143,44	10,13
OpenDSS modificado	147,46	-7,66

Fonte: Autor.

A segunda forma de corrigir foi aplicada e os resultados estão apresentados na 4.25, podendo destacar os valores de 147,46 MW e -7,66 Mvar. Percebe-se que a potência ativa teve uma diferença considerável com relação a simulação no ANAREDE, e a potência reativa além de ser diferente, possuem um sinal negativo, isto é, a barra de referência está absorvendo energia reativa da rede, ao invés de injetar como na simulação do ANAREDE indica.

Figura 4.25 – Resultado da simulação da rede de 9 barras com a correção no OpenDSS.



Fonte: Autor.

Com este segundo exemplo já é possível inferir com mais segurança que a conversão do cartão .pwf para .dss está funcionando corretamente. Os valores resultantes dos fluxos de potência em ambos os programas estão atingindo valores semelhantes. Nesse sistema elétrico, ficou ainda mais evidente que desde que os dados estejam coerentes o resultado da simulação serão parecidos, já que não foi preciso corrigir valores de potências reativas de geradores. E visando uma conversão com menos artificialidade, isto é, estipular valores para os equipamentos além dos dados que o cartão .pwf oferece, nota-se que a conversão é satisfatória.

É valido destacar que neste caso o cartão do ANAREDE possuía um "("escrito no início linha 33 do cartão, além de um dígito faltando no fim de cada linha de 32 até 41, ou seja, o cartão veio com alguns erros. Isto causou problemas na conversão, para tanto foi preciso realizar ajustes no cartão para que o algoritmo funcionasse corretamente.

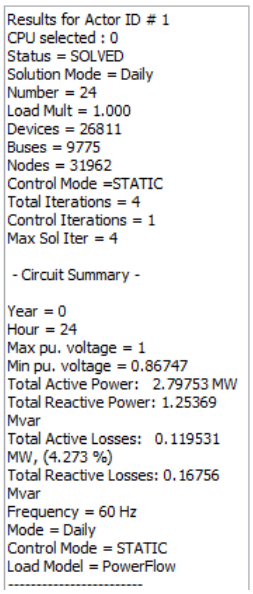
4.2 INTEGRAÇÃO ENTRE A SUBTRANSMISSÃO E A DISTRIBUIÇÃO

Uma vez que a conversão de um cartão do ANAREDE para o OpenDSS foi validada e apresenta valores satisfatórios para o fluxo de potência, a próxima etapa é converter o cartão .pwf de uma rede de transmissão e integrá-la com uma rede de distribuição de modo que seja possível realizar um fluxo de potência com as redes integradas.

4.2.1 Rede de Tramandaí

A rede de média e baixa tensão que será tomada para exemplo na metodologia de integração é um alimentador de Tramandaí, sendo que este está modelado no OpenDSS. Os resultados da simulação do fluxo de potência estão apresentados na Figura 4.26, podendo destacar os valores de potência ativa total, 2,80 MW e potência reativa total 1,25 Mvar.

Figura 4.26 – Resultado da simulação da rede de Tramandaí no OpenDSS.



Fonte: Autor.

4.2.2 Validação da conversão do Litoral Norte

A avaliação da conversão do cartão do ANAREDE para a sua representação no OpenDSS é desenvolvida comparando valores de fluxo de potência obtidos nos dois programas e caso estes valores forem semelhantes, há boas razões para apontar que a conversão foi realizada de maneira adequada. Por isso, inicialmente , o cartão da rede do Litoral Norte do ANAREDE, na Figura 4.27, é convertido para o formato da modelagem do OpenDSS, Figura 4.28. Nas Figuras 4.29 e 4.30 é constata-se que os valores do fluxo de potência, em especial da barra $V\theta$ do ANAREDE e da potência ativa e reativas totais da simulação do OpenDSS correspondem.

Figura 4.27 – Cartão da rede Litoral Norte do ANAREDE.

```
TITU
Litoral_Norte
DBAR
(Num)OETGb(    nome    )Gl( V) ( A) ( Pg) ( Qg) ( Qn) ( Qm)(Bc    ) ( Pl) ( Ql) ( Sh)Are(Vf)M(1) (2) (3) (4) (5) (6) (7) (8) (9) (10
     1 L269Atlantic-69  51020-58.   0.-.119-999999999                          11000
50048 L 69ENT936----69  1020-58.                                               11000
50052 L 69ENT804C---69  1020-58.                                               11000
50058 L 69ENT804B---69  1020-58.                                               11000
50066 L 69EN804A---69   1020-58.                                               11000
99999
DLIN
(De )d O d(Pa )NcEP  ( R% ) ( X% ) (Mvar) (Tap) (Tmn) (Tmx) (Phs) (Bc    ) (Cn) (Ce)Ns(Cq) (1) (2) (3) (4) (5) (6) (7) (8) (9) (10
     1     50048 1     3.5510.317  .175                            75. 90.   75.
     1     50052 1     .372  1.06  .018                            75. 90.   75.
     1     50058 1     .372  1.06  .018                            75. 90.   75.
     1     50066 1     .372  1.06  .018                            75. 90.   75.
99999
DGLT
(G (Vmn)  (Vmx)  (Vmne  (Vmxe
 5   .95  1.05    .95  1.05
 0   .95  1.05    .95 1.048
99999
DARE
(Ar     (Xchg)      (      Identificacao da area       ) (Xmin)  (Xmax)
  1         0.                                             0.  3000.
99999
DGBT
(G ( kV)
69   69.
99999
FIM
```

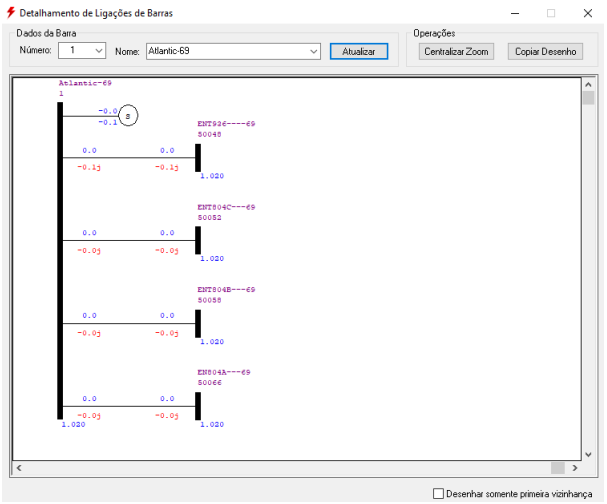
Fonte: Autor.

Figura 4.28 – Cartão da rede Litoral Norte convertido para OpenDSS.

```
Clear

// ------------------------------ Novo Circuito --------------------------------

New circuit.Atlantic-69 bus1=1 baseMVA=100.0 basekv=69. angle=-58.0 pu=1.02 phases=3 frequency=60

// -------------------------- Linecode -------------------------------

New Linecode.Cabo0 R1=1.69015 X1=4.91192 Units=km

New Linecode.Cabo1 R1=0.177109 X1=0.504666 Units=km

New Linecode.Cabo2 R1=0.177109 X1=0.504666 Units=km

New Linecode.Cabo3 R1=0.177109 X1=0.504666 Units=km

// ---------------------------- Linhas ---------------------------------------------

New Line.L0 bus1=1 bus2=50048 linecode=Cabo0 length=1 units=km

New Line.L1 bus1=1 bus2=50052 linecode=Cabo1 length=1 units=km

New Line.L2 bus1=1 bus2=50058 linecode=Cabo2 length=1 units=km

New Line.L3 bus1=1 bus2=50066 linecode=Cabo3 length=1 units=km

// ---------------------------- Capacitor ---------------------------------------------

New capacitor.CAP0 bus1=1 kv=69. kvar=175.0 phases=3

New capacitor.CAP1 bus1=1 kv=69. kvar=18.0 phases=3

New capacitor.CAP2 bus1=1 kv=69. kvar=18.0 phases=3

New capacitor.CAP3 bus1=1 kv=69. kvar=18.0 phases=3

// ------------------------------------ Tipo de Solução -------------------------

Set mode=snapshot

// ------------------------------- Valores bases de tensão -------------------------

Set voltagebases=[ 69.]
calcvoltagebases

// ------------------------------------ Solucionar -------------------------

Solve
```

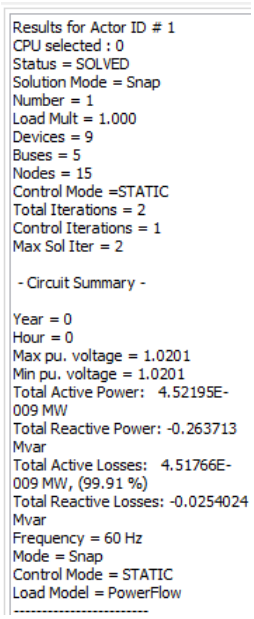
Fonte: Autor.

Figura 4.29 – Simulação do Litoral Norte sem carga no ANAREDE.



Fonte: Autor.

Figura 4.30 – Simulação do Litoral Norte sem carga no OpenDSS.



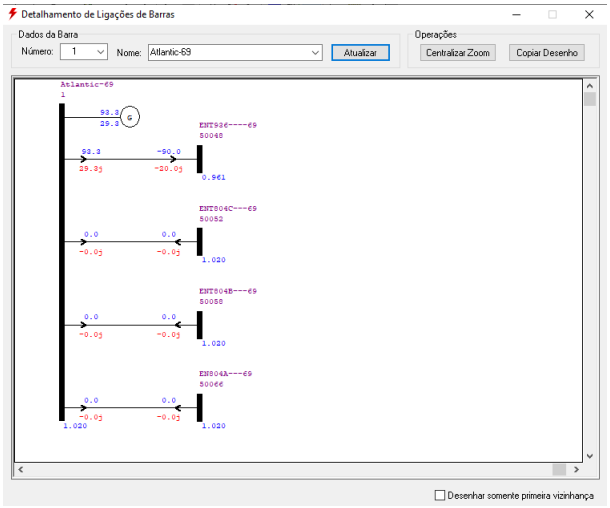
Fonte: Autor.

4.2.3 Litoral Norte com uma carga e com três cargas

Para avaliar o comportamento da rede Litoral Norte, serão simulados dois casos adicionais: um com um carga concentrada em uma barra, e outro com cargas em três barras. No primeiro caso adicional, acrescenta-se na rede Litoral Norte uma carga, nesse

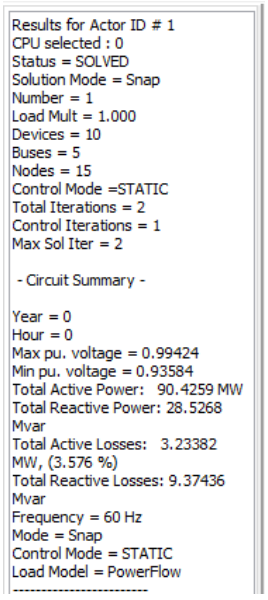
caso de 90 MW e 20 Mvar, em uma das barras no arquivo .pwf, depois converte-se o arquivo, simula-se o fluxo de potência com OpenDSS e compara-se com a simulação do ANAREDE. Então, no segundo caso adicional, um processo análogo é elaborado, porém com três cargas na rede, a saber, de 10 MW e 2 Mvar, 5 MW e 1 Mvar e, por fim, 6 MW e 1 Mvar . Os resultados das simulações são indicados nas Figuras 4.31, 4.32, 4.33 e 4.34.

Figura 4.31 – Simulação do Litoral Norte com uma carga no ANAREDE.



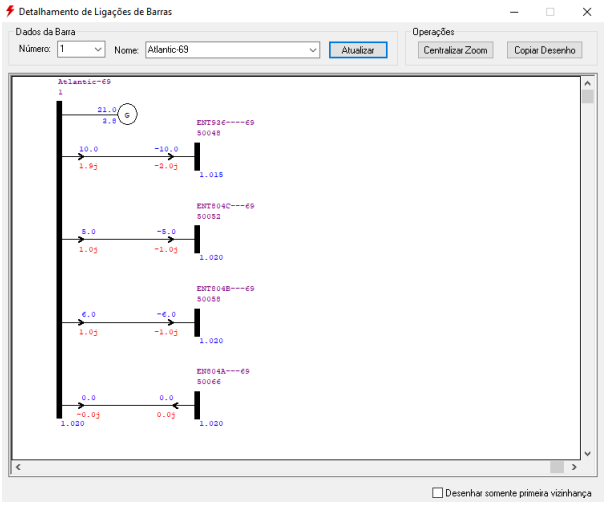
Fonte: Autor.

Figura 4.32 – Simulação do Litoral Norte com uma carga no OpenDSS.



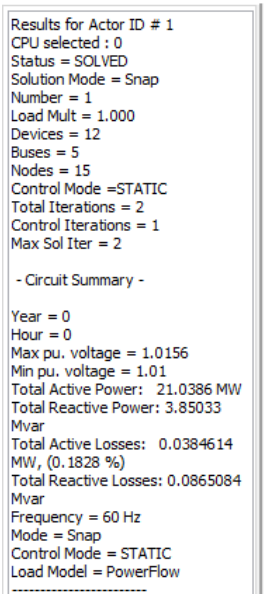
Fonte: Autor.

Figura 4.33 – Simulação do Litoral Norte com três cargas no ANAREDE.



Fonte: Autor.

Figura 4.34 – Simulação do Litoral Norte com três cargas no OpenDSS.



Fonte: Autor.

4.2.4 Processo de integração das redes

A das redes de transmissão e distribuição é elaborada alterando o arquivo "circuit" empregado pelo código mestre da rede de Tramandaí no OpenDSS. O arquivo "'CircuitoMT_ 5707_TRA_7_——1——.dss'" é alterado para que o "circuit" agora seja lido como a rede do Litoral Norte. Vale destacar que para realizar a integração foi necessário adicionar

um transformador e renomear as barras para que a integração funcione corretamente. Nas Figuras 4.35 e 4.36 estão mostrados os códigos do "circuit" antes e depois da modificação, respectivamente.

Figura 4.35 – Arquivo do "circuit"original.

```
! Criação da seção de circuitos
New "Circuit.TRA_7" basekv=13.8 pu=1 bus1="53934253A" r1=0 x1=0.0001
```

Fonte: Autor.

Figura 4.36 – Arquivo do "circuit" alterado.

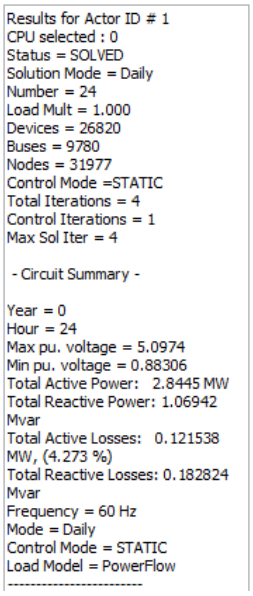
```
Clear

// ------------------------------- Novo Circuito ---------------------------------

New circuit.Atlantic-69 bus1=1 baseMVA=100.0 basekv=69. angle=-58.0 pu=1.02 phases=3 frequency=60

// ------------------------------ Transformador --------------------------

New transformer.TR0 XHL=15. %r=0.05 numtaps=0 windings=2
~ wdg=1 bus=1 tap=1.0 conn=wye kv=69 kva=100000
~ wdg=2 bus=53934253A conn=wye kv=13.8 kva=100000
```

Fonte: Autor.

4.2.5 Resultados das redes integradas

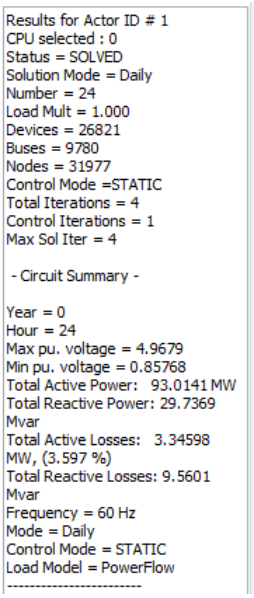
Com respeito a integração da rede Litoral Norte com a rede de Tramandaí haverá três situações para serem discutidos. O primeiro caso é com a rede do Litoral Norte sem cargas, o segundo caso é com a rede do Litoral Norte com uma carga e o terceiro caso é com a rede do Litoral Norte com três cargas. Os resultados são apresentados nas Figuras 4.37, 4.38 e 4.39.

Figura 4.37 – Simulação do Litoral Norte sem cargas integrado a rede de Tramandaí no OpenDSS.



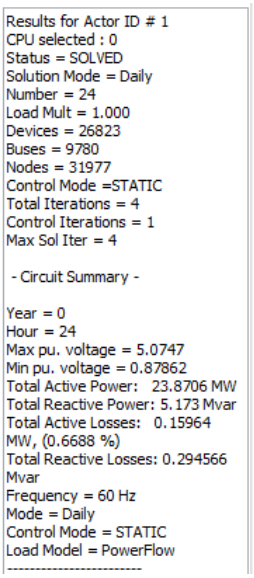
Fonte: Autor.

Figura 4.38 – Simulação do Litoral Norte com uma carga integrado a rede de Tramandaí no OpenDSS.



Fonte: Autor.

Figura 4.39 – Simulação do Litoral Norte com 3 cargas integrado a rede de Tramandaí no OpenDSS.



Fonte: Autor.

4.2.6 Discussão dos resultados

Os três casos da rede Litoral Norte integrada a rede de Tramandaí são apresentados na Tabela 4.3, juntamente com os casos antes da integração. Ali são indicados os valores do "circuit", com respeito ao potência ativa e reativo após o fluxo de potência. Na tabela, abrevia-se Litoral Norte por LN.

Tabela 4.3 – Resultados das simulações da Rede Litoral Norte integrada com Tramandaí.

Caso	Pot. Ativa (MW)	Pot. Reativa (Mvar)
Rede Tramandaí	2,80	1,25
LN sem carga	0	-0,26
LN com 1 carga	90,43	28,53
LN com 3 cargas	21,04	3,85
Primeiro caso	2,84	1,07
Segundo caso	93,01	29,74
Terceiro caso	23,87	5,17

Fonte: Autor.

Percebe-se que ocorre uma soma das potências ativas e reativas, com relação as potências na rede de Tramandaí e a rede Litoral Norte. No primeiro caso é possível

visualizar que a potência reativa total é alterada, indo para 1,07 Mvar dado que havia 1,25 Mvar na rede de Tramandaí e -0,26 Mvar na rede do Litoral Norte. No segundo caso foi preciso demandar mais potência do "circuit", no caso 93,01 MW, uma vez que a carga há a carga integrada com a rede de Tramandaí que já solicitava 2,80MW. No terceiro caso, verifica-se algo parecido com os outros casos, onde o valor de potência ativa total bem como o valor de potência reativa um pouco maiores.

Diante desses resultados pode-se inferir que o trânsito de potência entre as redes está se sucedendo, o que indicaria que as redes estão integradas. Mesmo assim, nota-se uma diferença nos valores das potências somadas comparando elas separadas e integradas, algo que pode estar associada a própria modelagem das redes de distribuição e transmissão, tendo em vista que a relação X/R é distinta nesses dois âmbitos da rede, o que pode implicar em simplificações nos cálculos, que por sua vez explicariam os pequenos erros nos resultados.

Nota-se ainda que o valor "max PU voltage" assumiu um valor em torno de 5 p.u nas simulações. A razão disto é que o OpenDSS está encontrando um valor perto de 69 kV para o alimentador, porém numa base de 13,8 kV. Como 69 é 5 vezes maior que 13,8 este valor em torno de 5 p.u. é explicado. A decisão de não alterar nas bases de cálculo foi pensada tendo em vista que as modificações sejam mínimas no arquivo da rede de Tramandaí nesta integração.

5 CONCLUSÃO

O sistema elétrico brasileiro possui uma complexidade e vastidão elevadas e visando executar diagnósticos, análises e planejamento é preciso utilizar ferramentas computacionais avançadas. Como organizações podem empreender programas distintos, as análises dos diferentes âmbitos no qual a energia perpassa podem ser dificultadas. A exemplo, tem-se o ANAREDE para a transmissão de energia, enquanto se usa o OpenDSS como software para a distribuição de energia.

Este trabalho procurou desenvolver uma metodologia onde seria possível realizar análises de maneira integrada entre as redes de distribuição e transmissão no programa OpenDSS. Por isso, foram realizadas duas etapas, uma de validação da conversão do cartão do ANARADE para OpenDSS e outra na qual se aplicou a metodologia de integração em uma rede real.

Desejou-se obter uma conversão precisa do cartão do ANAREDE para OpenDSS de maneira automatizada, para tanto foi necessário criar uma ferramenta no ambiente Python que pudesse realizar a conversão dos cartões provenientes do ANAREDE para um formato no qual o OpenDSS poderia ler. O objetivo era obter a rede elétrica de interesse de maneira mais fidedigna ao cartão em .pwf, evitando adicionar artificialmente dados ou perder informações relevantes. Assim teríamos a rede modelada e lida pelo ANAREDE, sendo representada adequadamente no OpenDSS.

Ao analisar os resultados do fluxo de potência comparando os valores obtidos no ANAREDE e no OpenDSS para o Sistema de 5 barras, bem como o Sistema de 9 barras, é possível constatar que as valores são próximos, podendo inferir que a conversão das redes está ocorrendo de maneira adequada com respeito ao fluxo de potência.

Diante da análise dos resultados do estudo do caso real, a rede Litoral Norte com a rede de Tramandaí, constata-se que a integração das redes de fato está ocorrendo de modo satisfatório. Isto implica na possibilidade de integrar outras redes, bem como na exploração dessa metodologia de integração como um recurso para planejamento e análise de sistemas de potência que envolvem distribuição e transmissão, no software OpenDSS.

Dentre as possibilidades de aplicação dessa metodologia pode-se incluir a análise de geração de distribuída, que vêm crescendo nos últimos anos (MIRANDA, 2013), e tem sido realizada no OpenDss, de modo que poderia ser levado em conta o impacto na rede de transmissão. Outra situação de interesse é a análise de Montantes de Uso de Sistema de Transmissão (MUST), isto é, a quantidade de energia que cada distribuidora deve contratar para suprir sua demanda em um determinado tempo. Além disso, esta ferramenta pode ser levada em conta no planejamento de novas redes de distribuição, para saber o impacto que será causado na transmissão de energia.

Com respeito a trabalhos futuros, sugere-se que outros códigos de execução do

ANAREDE possam ser convertidos para o uso no ambiente Pyhton e que aquelas informações que OpenDSS puder fazer uso sejam convertidas. Seria interessante explorar o caso da conversão de geradores e procurar formas de melhorar sua representação, além de explorar outros tipos de equipamentos como transformador de 3 enrolamentos. Outra recomendação seria aplicar em mais exemplos do ANAREDE, com diferentes configurações e tamanhos da rede, para validar a confiabilidade dos resultados da conversão. Além disso, pode-se aplicar a metodologia proposta nos casos de análise de impacto de GD ou MUST.

Pode-se dizer portanto, que este trabalho cumpriu adequadamente sua proposta, uma vez que o objetivo geral de desenvolver a metodologia de análise integrada no OpenDSS foi realizado, além disso pôde-se cumprir os objetivos específicos, dado que se compreendeu melhor as redes elétricas, a linguagem Python e os programas ANAREDE e OpenDSS, e aplicou-se a alguns casos e, por fim, analisou-se os resultados, que apresentaram valores satisfatórios.

REFERÊNCIAS BIBLIOGRÁFICAS

ANAREDE. **Programa de Análise de Redes: Manual do Usuário. V11.00.01**. Rio de Janeiro: DRE, 2019.

ANDRADE, V. B. et al. Modelagem de um sistema de distribuição real desbalanceado e análise do impacto da geração distribuída utilizando o software opendss. **Sociedade Brasileira de Automática**, 2020.

BANIN, S. L. **Python 3 : conceitos e aplicações : uma abordagem didática**. 1. ed. São Paulo: Érica, 2018.

BARROS, B. F. de; BORELLI, R.; GEDRA, R. L. **Geracao, transmissao, distribuicao e consumo de energia eletrica**. 1. ed. São Paulo: Érica, 2014.

COSTER, E. J. et al. Integration issues of distributed generation in distribution grids. **Proceedings of the IEEE**, 2011.

DUGAN, R. C.; MONTENEGRO, D. **Reference Guide: The Open Distribution System Simulator**^{TM}. Electric Power Research Institute, Inc: OpenDSS, 2020.

GIL, A. C. **Como elaborar projetos de pesquisa**. 6. ed. São Paulo: Atlas, 2018.

HALLIDAY, D.; RESNICK, R.; WALKER, J. **Fundamentos de física, volume 3 : eletromagnetismo**. 10. ed. Rio de Janeiro: LTC, 2020.

MIRANDA, R. F. C. **Análise da Inserção de Geração Distribuída de Energia Solar Fotovoltaica no Setor Residencial Brasileiro**. 2013. 291 f. Monografia (Dissertação de Mestrado) — Programa de Pós-Graduação em Planejamento Energético, COPPE, Universidade Federal de Rio de Janeiro, Rio de Janeiro, 2013.

MOHAN, N. **Sistemas elétricos de potência: curso introdutório**. 1. ed. Rio de Janeiro: LTC, 2016. Tradução de Walter Denis Cruz Sanchez.

OLIVEIRA, C. B. de et al. **Introdução a sistemas elétricos de potência componentes simétricas**. 2. ed. São Paulo: Blucher, 2000.

ONS. **Submódulo 18.2: Relação dos sistemas e modelos computacionais**. Brasil: Operador Nacional do Sistema, 2008.

ONS. **Sobre o SIN: Sistemas Isolados**. Operador Nacional do Sistema, 2022. Acesso em 17 mai. 2022. Disponível em: <http://www.ons.org.br/paginas/sobre-o-sin/sistemas-isolados>.

PERKOVIC, L. **Introdução à computação usando Python : um foco no desenvolvimento de aplicações**. 1. ed. Rio de Janeiro: LTC, 2016.

PINTO, M. de O. **Energia elétrica : geração, transmissão e sistemas interligados**. 1. ed. Rio de Janeiro: LTC, 2018.

ROBBA, E. J. et al. **Análise de sistemas de transmissão de energia elétrica**. 1. ed. São Paulo: Blucher, 2020.

SAMPIERI, H.; COLLADO, C. F.; LUCIO, M. del P. B. **Metodologia de pesquisa**. 5. ed. Porto Alegre: Penso, 2013.

SILVEIRA, M. F. da et al. **Sistemas elétricos de potência**. 1. ed. Porto Alegre: SAGAH, 2021.

SORDI, J. O. D. **Desenvolvimento de projeto de pesquisa**. 1. ed. São Paulo: Saraiva, 2017.

SOUZA, J. P. A. de; FERREIRA, J. P. K.; CAMPOS, G. E. M. Uso do opendss em aplicações reais para distribuidoras de energia. **Sociedade Brasileira de Automática**, 2020.

WULF W. M. A. **Great Achievements and Grand Challenges**. National Academy of Engineering, 2000. Acesso em 06 jan. 2022. Disponível em: <https://www.nae.edu/7461/GreatAchievementsandGrandChallenges>.

ANEXO A – CÓDIGOS

A.1 – CONVERSOR_PWF_PARA_PY

```
import numpy as np
import math

#Definir qual arquivo será convertido
nome = list(open("SIST5BARRAS.PWF","r"))
nome = [s.rstrip() for s in nome]
A = np.array(nome)
#print(A)
n=len(A)
#print(n)
B=A.T
#print(B)

#Separar cada termo
i=0
C=[]
E=[]
while(i<n):
    C = B[i]
    m=len(C)
    j=0
    linhaE = []
    while (j<m):
        linhaE.append(C[j])
        j = j + 1
    E.append(linhaE)
    i = i + 1
#print(E)

#Escrevendo DBAR

K=[]            #vetor para receber o DBAR
i=              #linha no pwf para começar a conversão
```

```
while(i<):       #linha no pwf para terminar a conversão
    t=len(E[i])
    #print(t)
    # Número
    K1 = []
    N = [E[i][0], E[i][1], E[i][2], E[i][3], E[i][4]]
    #print(N)
    m = len(N)
    j = 0
    while (j < m):
        if N[j] != '␣':
            K1.append(N[j])
        j = j + 1
        Kr1= ''.join(K1)
    #print(Kr1)

    # Operação
    K2 = []
    if E[i][5] == '␣':
        K2.append('0')
    else:
        K2.append(E[i][5])
    #print(K2)

    # Estado
    K3 = []
    if E[i][6] == '␣':
        K3.append('0')
    else:
        K3.append(E[i][6])
    #print(K3)

    # Tipo
    K4 = []
    if E[i][7] == '␣':
        K4.append('0')
    else:
        K4.append(E[i][7])
    #print(K4)
```

```
# Grupo Base de Tensão
K5 = []
GBT = [E[i][8], E[i][9]]
# print(GBT)
m = len(GBT)
j = 0
while (j < m):
    if GBT[j] != '␣':
        K5.append(GBT[j])
        #print(K5)
    j = j + 1
    Kr5= ''.join(K5)
    if len(K5) == 0:
        Kr5='0'
#print(Kr5)
#print(K5)

# Nome
K6 = []
Nome = [E[i][10], E[i][11], E[i][12], E[i][13], E[i][14],
        E[i][15], E[i][16], E[i][17], E[i][18], E[i][19],
        E[i][20], E[i][21]]
#print(Nome)
m=len(Nome)
j = 0
while (j < m):
    if Nome[j] != '␣':
        K6.append(Nome[j])
    j = j + 1
    Kr6= ''.join(K6)
#print(Kr6)

# Grupo Limite de Tensão
K7 = []
GLT = [E[i][22], E[i][23]]
# print(GLT)
m = len(GLT)
j = 0
```

```
while (j < m):
    if GLT[j] != '␣':
        K7.append(GLT[j])
    j = j + 1
    Kr7= ''.join(K7)
if len(K7) == 0:
    Kr7='0'

# Tensão

K8 = []
T = [E[i][24], E[i][25], E[i][26], E[i][27]]
# print(T)
m = len(T)
j = 0
while (j < m):
    if T[j] != '␣':
        K8.append(T[j])
    j = j + 1
    Kr8= ''.join(K8)
if len(K8) == 0:
    Kr8='0.'
#print(Kr8)

# Ângulo

K9 = []
Ang = [E[i][28], E[i][29], E[i][30], E[i][31]]
# print(Ang)
m = len(Ang)
j = 0
while (j < m):
    if Ang[j] != '␣':
        K9.append(Ang[j])
    j = j + 1
    Kr9= ''.join(K9)
if len(K8) == 0:
    Kr9='0.'
#print(Kr9)
```

```
# Geração Ativa
K10 = []
Gat = [E[i][32], E[i][33], E[i][34], E[i][35], E[i][36]]
# print(Gat)
m = len(Gat)
j = 0
while (j < m):
    if Gat[j] != '␣':
        K10.append(Gat[j])
    j = j + 1
    Kr10= ''.join(K10)
if len(K10) == 0:
    Kr10='0.'
#print(Kr10)

# Geração Reativa
K11 = []
Grat = [E[i][37], E[i][38], E[i][39], E[i][40], E[i][41]]
m = len(Grat)
j = 0
while (j < m):
    if Grat[j] != '␣':
        K11.append(Grat[j])
    j = j + 1
    Kr11= ''.join(K11)
if len(K11) == 0:
    Kr11='0.'
#print(Kr11)

# Geração Reativa Mínima
K12 = []
Grmin = [E[i][42], E[i][43], E[i][44], E[i][45], E[i][46]]
# print(Grmin)
m = len(Grmin)
j = 0
while (j < m):
    if Grmin[j] != '␣':
        K12.append(Grmin[j])
```

```
        j = j + 1
        Kr12= ''.join(K12)
    if len(K12) == 0:
        Kr12='0.'
    #print(Kr12)

    # Geração Reativa Máxima
    K13 = []
    Grmax = [E[i][47], E[i][48], E[i][49], E[i][50], E[i][51]]
    # print(Grmax)
    m = len(Grmax)
    j = 0
    while (j < m):
        if Grmax[j] != ' ':
            K13.append(Grmax[j])
        j = j + 1
        Kr13= ''.join(K13)
    if len(K13) == 0:
        Kr13='0.'
    #print(Kr13)

    # Barra Controlada
    K14 = []
    Bctrl = [E[i][52], E[i][53], E[i][54], E[i][55], E[i][56],
            E[i][57]]
    # print(Bctrl)
    m = len(Bctrl)
    j = 0
    while (j < m):
        if Bctrl[j] != ' ':
            K14.append(Bctrl[j])
        j = j + 1
        Kr14= ''.join(K14)
    if len(K14) == 0:
        Kr14='0.'
    #print(Kr14)

    # Carga Ativa
    K15 = []
```

```
Cat = [E[i][58], E[i][59], E[i][60], E[i][61], E[i][62]]
# print(Cat)
m = len(Cat)
j = 0
while (j < m):
    if Cat[j] != '␣':
        K15.append(Cat[j])
    j = j + 1
    Kr15= ''.join(K15)
if len(K15) == 0:
    Kr15='0.'
#print(Kr15)

# Carga Reativa
K16 = []
Crat = [E[i][63], E[i][64], E[i][65], E[i][66], E[i][67]]
# print(Crat)
m = len(Crat)
j = 0
while (j < m):
    if Crat[j] != '␣':
        K16.append(Crat[j])
    j = j + 1
    Kr16= ''.join(K16)
if len(K16) == 0:
    Kr16='0.'
#print(Kr16)

if t == 68:
    Kr17='0'
else:
    # Capacitor Reator
    K17 = []
    Cp = [E[i][68], E[i][69], E[i][70], E[i][71], E[i][72]]
    # print(Cp)
    m = len(Cp)
    j = 0
    while (j < m):
        if Cp[j] != '␣':
```

```
            K17.append(Cp[j])
        j = j + 1
        Kr17 = ''.join(K17)
    if len(K17) == 0:
        Kr17 = '0.'
    # print(Kr17)

Kr18 = '0'
Kr19 = '0'

if t > 73:
    # Área
    K18 = []
    Area = [E[i][73], E[i][74], E[i][75]]
    # print(Area)
    m = len(Area)
    j = 0
    while (j < m):
        if Area[j] != '␣':
            K18.append(Area[j])
        j = j + 1
        Kr18 = ''.join(K18)
    if len(K18) == 0:
        Kr18 = '0.'
    # print(Kr18)

if t > 76:
    # Tensão para definição de Carga
    K19 = []
    Tencar = [E[i][76], E[i][77], E[i][78], E[i][79]]
    # print(Tencar)
    m = len(Tencar)
    j = 0
    while (j < m):
        if Tencar[j] != '␣':
            K19.append(Tencar[j])
        j = j + 1
        Kr19 = ''.join(K19)
    if len(K19) == 0:
```

```
            Kr19 = '0.'
        # print(Kr19)

    Klinha = [Kr1, K2[0], K3[0], K4[0], Kr5, Kr6, Kr7, Kr8, Kr9,
    Kr10, Kr11, Kr12, Kr13, Kr14,Kr15, Kr16, Kr17, Kr18, Kr19]
    #print(Klinha)

    K.append(Klinha)
    i=i+1

DBAR=K
print(DBAR)

##Escrevendo DLIN

K=[]              #vetor para receber o DLIN
i=                #linha no pwf para começar a conversão
while(i<):        #linha no pwf para terminar a conversão
    t=len(E[i])
    #print(t)
    # Da barra
    K1 = []
    DBarra = [E[i][0], E[i][1], E[i][2], E[i][3], E[i][4]]
    #print(N)
    m = len(DBarra)
    j = 0
    while (j < m):
        if DBarra[j] != '␣':
            K1.append(DBarra[j])
        j = j + 1
        Kr1= ''.join(K1)
    #print(Kr1)

    # Abertura da Barra
    K2 = []
    if E[i][5] == '␣':
        K2.append('L')
    else:
        K2.append(E[i][5])
```

```
#print(K2)

# Nenhum 07
K3 = ['0']
#print(K3)

# Operação
K4 = []
if E[i][7] == '␣':
    K4.append('A')
else:
    K4.append(E[i][7])
#print(K4)

# Nenhum 09
K5 = ['0']
#print(K5)

# Abertura para Barra
K6 = []
if E[i][9] == '␣':
    K6.append('L')
else:
    K6.append(E[i][9])
#print(K6)

# Para Barra
K7 = []
PBarra = [E[i][10], E[i][11], E[i][12],E[i][13],E[i][14]]
# print(PBarra)
m = len(PBarra)
j = 0
while (j < m):
    if PBarra[j] != '␣':
        K7.append(PBarra[j])
    j = j + 1
    Kr7= ''.join(K7)
    if len(K7) == 0:
        Kr7='0'
```

```
#print(Kr7)
#print(K7)

#Circuito
K8 = []
Circ = [E[i][15], E[i][16]]
# print(Circ)
m = len(Circ)
j = 0
while (j < m):
    if Circ[j] != '␣':
        K8.append(Circ[j])
    j = j + 1
    Kr8= ''.join(K8)
    if len(K8) == 0:
        Kr8='0'
#print(Kr8)

#Estado
K9 = []
if E[i][17] == '␣':
    K9.append('L')
else:
    K9.append(E[i][17])
#print(K9)

#Proprietário
K10 = []
if E[i][18] == '␣':
    K10.append('F')
else:
    K10.append(E[i][18])
#print(K10)

#Nenhum 20
K11 = ['0']
#print(K11)

#Resistência
```

```
K12 = []
R = [E[i][20], E[i][21], E[i][22],E[i][23],E[i][24],E[i][25]]
# print(R)
m = len(R)
j = 0
while (j < m):
    if R[j] != '␣':
        K12.append(R[j])
    j = j + 1
    Kr12= ''.join(K12)
    if len(K12) == 0:
        Kr12='0'
#print(Kr12)
#print(K12)

#Reatância
K13 = []
X = [E[i][26], E[i][27], E[i][28],E[i][29],E[i][30],E[i][31]]
# print(X)
m = len(X)
j = 0
while (j < m):
    if X[j] != '␣':
        K13.append(X[j])
    j = j + 1
    Kr13= ''.join(K13)
    if len(K13) == 0:
        Kr13='0'
#print(Kr13)
#print(K13)

Kr14 = '0'
Kr15 = '0'
Kr16 = '0'
Kr17 = '0'
Kr18 = '0'
Kr19 = '0'
Kr20 = '0'
Kr21 = '0'
```

```
Kr22 = '0'
Kr23 = '0'

if t > 32:
    # Susceptância
    K14 = []
    S = [E[i][32], E[i][33], E[i][34], E[i][35], E[i][36],
        E[i][37]]
    # print(S)
    m = len(S)
    j = 0
    while (j < m):
        if S[j] != '␣':
            K14.append(S[j])
        j = j + 1
        Kr14 = ''.join(K14)
        if len(K14) == 0:
            Kr14 = '0'
    # print(Kr14)
    # print(K14)

if t > 38:
    # Tap
    K15 = []
    Tap = [E[i][38], E[i][39], E[i][40], E[i][41], E[i][42]]
    # print(Tap)
    m = len(Tap)
    j = 0
    while (j < m):
        if Tap[j] != '␣':
            K15.append(Tap[j])
        j = j + 1
        Kr15 = ''.join(K15)
        if len(K15) == 0:
            Kr15 = '0'
    # print(Kr15)
    # print(K15)

if t > 43:
```

```
        # Tap mínimo
        K16 = []
        Tapmin = [E[i][43], E[i][44], E[i][45], E[i][46], E[i][47]]
        # print(Tapmin)
        m = len(Tapmin)
        j = 0
        while (j < m):
            if Tapmin[j] != '␣':
                K16.append(Tapmin[j])
            j = j + 1
            Kr16 = ''.join(K16)
            if len(K16) == 0:
                Kr16 = '0'
        # print(Kr16)
        # print(K16)

    if t > 48:
        # Tap máximo
        K17 = []
        Tapmax = [E[i][48], E[i][49], E[i][50], E[i][51], E[i][52]]
        # print(Tapmax)
        m = len(Tapmax)
        j = 0
        while (j < m):
            if Tapmax[j] != '␣':
                K17.append(Tapmax[j])
            j = j + 1
            Kr17 = ''.join(K17)
            if len(K17) == 0:
                Kr17 = '0'
        # print(Kr17)
        # print(K17)

    if t > 53:
        # Defasagem
        K18 = []
        Dfsg = [E[i][53], E[i][54], E[i][55], E[i][56], E[i][57]]
        # print(Dfsg)
        m = len(Dfsg)
```

```
        j = 0
        while (j < m):
            if Dfsg[j] != '␣':
                K18.append(Dfsg[j])
            j = j + 1
            Kr18 = ''.join(K18)
            if len(K18) == 0:
                Kr18 = '0'
        # print(Kr18)
        # print(K18)

    if t > 58:
        # Barra controlada
        K19 = []
        Bctrlv = [E[i][58], E[i][59], E[i][60],
        E[i][61], E[i][62], E[i][63]]
        # print(Bctrlv)
        m = len(Bctrlv)
        j = 0
        while (j < m):
            if Bctrlv[j] != '␣':
                K19.append(Bctrlv[j])
            j = j + 1
            Kr19 = ''.join(K19)
            if len(K19) == 0:
                Kr19 = '0'
        # print(Kr19)
        # print(K19)

    if t > 64:
        # Capacidade Normal
        K20 = []
        Cnorm = [E[i][64], E[i][65], E[i][66], E[i][67]]
        # print(Cnorm)
        m = len(Cnorm)
        j = 0
        while (j < m):
            if Cnorm[j] != '␣':
                K20.append(Cnorm[j])
```

```
            j = j + 1
            Kr20 = ''.join(K20)
            if len(K20) == 0:
                Kr20 = '0'
        # print(Kr20)
        # print(K20)

    if t > 68:
        # Capacidade em Emergência
        K21 = []
        Cemer = [E[i][68], E[i][69], E[i][70],E[i][71]]
        # print(Cemer)
        m = len(Cemer)
        j = 0
        while (j < m):
            if Cemer[j] != '␣':
                K21.append(Cemer[j])
            j = j + 1
            Kr21 = ''.join(K21)
            if len(K21) == 0:
                Kr21 = '0'
        # print(Kr21)
        # print(K21)

    if t > 72:
        # Número de taps
        K22 = []
        Ntap = [E[i][72], E[i][73]]
        # print(Ntap)
        m = len(Ntap)
        j = 0
        while (j < m):
            if Ntap[j] != '␣':
                K22.append(Ntap[j])
            j = j + 1
            Kr22 = ''.join(K22)
            if len(K22) == 0:
                Kr22 = '0'
        # print(Kr22)
```

```
        # print(K22)

    if t > 74:
        # Capacidade do equipamento
        K23 = []
        Cequi = [E[i][74], E[i][75], E[i][76], E[i][77]]
        # print(Cequi)
        m = len(Cequi)
        j = 0
        while (j < m):
            if Cequi[j] != '␣':
                K23.append(Cequi[j])
            j = j + 1
            Kr23 = ''.join(K23)
            if len(K23) == 0:
                Kr23 = '0'
        # print(Kr23)
        # print(K23)

    Klinha = [Kr1, K2[0], K4[0], K6[0], Kr7, Kr8, K9[0], K10[0],
    Kr12, Kr13, Kr14,Kr15, Kr16, Kr17, Kr18, Kr19, Kr20, Kr21,
    Kr22, Kr23]
    K.append(Klinha)
    i = i + 1
    # print(Klinha)

DLIN=K
print(DLIN)

#Escrevendo DGBT

K=[]            #vetor para receber o DGBT
i=              #linha no pwf para começar a conversão
while(i<):      #linha no pwf para terminar a conversão
    t=len(E[i])
    #print(t)
    # Grupo base de tensão
    K1 = []
    GB = [E[i][0], E[i][1]]
```

```
        #print(N)
        m = len(GB)
        j = 0
        while (j < m):
            if GB[j] != '␣':
                K1.append(GB[j])
            j = j + 1
            Kr1= ''.join(K1)
            if len(Kr1) == 0:
                Kr1 = '0'
        #print(Kr1)

        # Tensão
        K2 = []
        Tas = [E[i][3], E[i][4], E[i][5], E[i][6], E[i][7]]
        #print(N)
        m = len(Tas)
        j = 0
        while (j < m):
            if Tas[j] != '␣':
                K2.append(Tas[j])
            j = j + 1
            Kr2= ''.join(K2)
            if len(Kr2) == 0:
                Kr2 = '1'
        #print(Kr2)
        Klinha = [Kr1, Kr2]
        #print(Klinha)
        K.append(Klinha)
        i=i+1

DGBT=K
print(DGBT)
```

A.2 – CONVERSOR_PY_PARA_DSS

```
import numpy as np
```

```
import math

# INSERINDO DBAR

DBAR = np.array()

# INSERINDO DLIN

DLIN = np.array()

# INSERINDO DCTE

DCTE =np.array([["Base", "100."]])

# INSERINDO DGBT

DGBT = np.array()

#BASE DE POTÊNCIA

if 'DCTE' in globals():
    Sbase=float(DCTE[0][1])*1E6
else:
    Sbase=100E6

#DEFININDO CIRCUIT

sizedbar=len(DBAR)
sizedlin=len(DLIN)
i = 0

while (i<sizedbar):
    if DBAR[i][3] == "2":
        if DBAR[i][2] == "L" or DBAR[i][2] == "0":
            linha = i
            i=sizedbar+1
    i=i+1

bus=DBAR[linha][0]
```

```
nome=DBAR[linha][5]
ang=float(DBAR[linha][8])
pu=float(DBAR[linha][7])/1000

# BASE DE TENSÃO DA BARRA VTHETA

sizedgbt = 0

if 'DGBT' in globals():
    sizedgbt = len(DGBT)
    i=0
    if DBAR[linha][4] == '0':
        base = '1'
    else:
        while (i < sizedgbt):
            if DBAR[linha][4] == DGBT[i][0]:
                base = DGBT[i][1]
            i = i + 1
else:
    base = '1'

#MATRIZ DE BASES DE TENSÃO

i=0
j=0
GB=[]

if 'DGBT' in globals():
    while (i < sizedgbt):
        j=0
        while (j < sizedbar):
            if DBAR[j][4] == DGBT[i][0]:
                linhaGB = []
                linhaGB.append(DBAR[j][0])
                linhaGB.append(DGBT[i][1])
                GB.append(linhaGB)
            elif DBAR[j][4] == '0':
                linhaGB = []
                linhaGB.append(DBAR[j][0])
```

```
                linhaGB.append('1')
                GB.append(linhaGB)
            j=j+1
        i = i + 1

else:
    while (j < sizedbar):
        DBAR[j][4] == '0'
        linhaGB = []
        linhaGB.append(DBAR[j][0])
        linhaGB.append('1')
        GB.append(linhaGB)
        j = j + 1

#TRAFO

i=0
j=0
traf=0
barradetr=[]
barraparatr=[]
XHL=[]
r=[]
tap=[]
numtap=[]
Strafo=[]
sizeGB=len(GB)
GB2=[]

while (i<sizedlin):
    if DLIN[i][11] != '0':
        if DLIN[i][6] == 'L' or DLIN[i][6] == '0':
            traf=1
            barradetr.append(DLIN[i][0])
            barraparatr.append(DLIN[i][4])
            base2=float(DLIN[i][4])
            k=0
            while(k<sizeGB):
                if base2 == float(GB[k][0]):
```

```
                    GB2.append(GB[k][1])
                k=k+1
            XHL.append(DLIN[i][9])
            r.append(float(DLIN[i][8]))
            tap.append(DLIN[i][11])
            numtap.append(DLIN[i][18])
            if DLIN[i][16] != '0':
                Strafo.append(float(DLIN[i][16])*1000)
            else:
                Strafo.append('100000')
            j=j+1
    i=i+1

if traf == 1:
    traf = len(barradetr)

#LINHA

i=0
lin=0
barrade=[]
barrapara=[]

while (i<sizedlin):
    if DLIN[i][11] == '0':
        if DLIN[i][6] == 'L' or DLIN[i][6] == '0':
            lin=1
            barrade.append(DLIN[i][0])
            barrapara.append(DLIN[i][4])
    i=i+1

if lin == 1:
    lin = len(barrade)


# LINECODE

i=0
j=0
```

```
k=0
linecode=0
cabo=[]
R1=[]
X1=[]

while (i<sizedlin):
    if DLIN[i][11] == '0':
        if DLIN[i][6] == 'L' or DLIN[i][6] == '0':
            linecode=1
            cabo.append(i)
            k=0
            while (k<sizedbar):
                if barrade[j]==GB[k][0]:
                    zbase = ((float(GB[k][1])*1000)**2)/(Sbase)/100
                    r1= float(DLIN[i][8]) * zbase
                    x1= float(DLIN[i][9])* zbase
                    R1.append(r1)
                    X1.append(x1)
                k=k+1
            j=j+1
    i=i+1

if linecode == 1:
    linecode = len(cabo)

# CARGA

i=0
cg=0
nomecarga=[]
barracarga=[]
carga=[]
cargar=[]
FPcar=[]

while (i<sizedbar):
    if DBAR[i][14] != '0' or DBAR[i][15] != '0':
        if DBAR[i][14] != '0.'or DBAR[i][15] != '0.':
```

```
            if DBAR[i][2] == 'L' or DBAR[i][2] == '0':
                cg = 1
                nomecarga.append(i)
                barracarga.append(DBAR[i][0])
                carga.append(float(DBAR[i][14]) * 1E3)
                cargar.append(float(DBAR[i][15]) * 1E3)
                if float(DBAR[i][15]) == 0.0:
                    FPcar.append('1')
                else:
                    FPcar.append(math.cos(math.atan((float(DBAR[i][15])
                    /float(DBAR[i][14])))))
    i = i + 1

linhascarga = len(barracarga)

if cg == 1:
    cg = len(barracarga)

#GERADOR

i = 0
ger=0
nomeger=[]
barrager=[]
Pger=[]
Qger=[]
FP=[]

while (i<sizedbar):
    if DBAR[i][3]=='1':
        if DBAR[i][2]=='L' or DBAR[i][2]=='0':
                ger=1
                nomeger.append(i)
                barrager.append((DBAR[i][0]))
                Qger.append(float(DBAR[i][10])*1E3)
                if float(DBAR[i][9]) == 0.0:
                    FP.append('1')
                    Pger.append('0.00001')
                else:
```

```
                    Pger.append(float(DBAR[i][9]) * 1E3)
                    FP.append(math.cos(math.atan((float(
                    DBAR[i][10])/float(DBAR[i][9])))))
    i = i+1

linhasger = len(barrager)

if ger == 1:
    ger = len(barrager)

#CAPACITOR

i = 0
j = 0
caps = 0
barradecap = []
mvar = []
GBcap = []

while (i<sizedlin):
    if DLIN[i][10] != '0':
        caps = 1
        barradecap.append(DLIN[i][0])
        GBcap.append(GB[i][1])
        mvar.append(float(DLIN[i][10])*1000)
    i = i + 1

if caps == 1:
    cap = len(barradecap)
else: cap=0

# CALCVOLTAGE BASES

cvbt = []

for i in range(len(GB)):
    cvbt.append(GB[i][1])
cvb = sorted(set(cvbt))
cvbn=len(cvb)
```

#DOCUMENTO

```
#arquivo = open("nbarras.dss","w")
arquivo = open("nbarras.txt","w")

arquivo.write("Clear \n\n")

arquivo.write("// ———————— Novo Circuito ———————— \n\n")

arquivo.write("New circuit.")
arquivo.write(nome)
arquivo.write(" bus1=")
arquivo.write(bus)
arquivo.write(" baseMVA=")
arquivo.write('{:.6}'.format(Sbase/1e6))
arquivo.write(" basekv=")
arquivo.write('{:.6}'.format(base))
arquivo.write(" angle=")
arquivo.write('{:.6}'.format(ang))
arquivo.write(" pu=")
arquivo.write('{:.6}'.format(pu))
arquivo.write(" phases=3 frequency=60 \n\n")

i=0
if traf>0:
    arquivo.write("// ———————— Transformador ———————— \n\n")
while (i<traf):
    arquivo.write("New transformer.TR")
    arquivo.write('{}'.format(i))
    arquivo.write(" XHL=")
    arquivo.write('{:.6}'.format(XHL[i]))
    arquivo.write(" %r=")
    arquivo.write('{:.6}'.format(r[i]/2))
    arquivo.write(" numtaps=")
    arquivo.write('{:.6}'.format(numtap[i]))
    arquivo.write(" windings=2 \n")
    arquivo.write("~ wdg=1 bus=")
    arquivo.write(barradetr[i])
```

```
    arquivo.write(" tap=")
    arquivo.write('{:.6}'.format(tap[i]))
    arquivo.write(" conn=wye kv=")
    arquivo.write(GB[i][1])
    arquivo.write(" kva=")
    arquivo.write('{:.6}'.format(Strafo[i]))
    arquivo.write(" \n")
    arquivo.write("~ wdg=2 bus=")
    arquivo.write(barraparatr[i])
    arquivo.write(" conn=wye kv=")
    arquivo.write(GB2[i])
    arquivo.write(" kva=")
    arquivo.write('{:.6}'.format(Strafo[i]))
    arquivo.write(" \n\n")
    i=i+1

i=0
if linecode>0:
    arquivo.write("// ---------- Linecode ---------- \n\n")
while (i<linecode):
    arquivo.write("New Linecode.Cabo")
    arquivo.write('{}'.format(cabo[i]))
    arquivo.write(" R1=")
    arquivo.write('{:.6}'.format(R1[i]))
    arquivo.write(" X1=")
    arquivo.write('{:.6}'.format(X1[i]))
    arquivo.write(" Units=km \n\n")
    i=i+1

i=0
if lin>0:
    arquivo.write("// ---------- Linhas ---------- \n\n")
while (i<lin):
    arquivo.write("New Line.L")
    arquivo.write('{}'.format(i))
    arquivo.write(" bus1=")
    arquivo.write(barrade[i])
    arquivo.write(" bus2=")
    arquivo.write(barrapara[i])
```

```
    arquivo.write(" linecode=Cabo")
    arquivo.write('{}'.format(cabo[i]))
    arquivo.write(" length=1 units=km \n\n")
    i=i+1

i=0
if cg>0:
    arquivo.write("// ---------- Cargas ---------- \n\n")
while (i<cg):
    arquivo.write("New load.C")
    arquivo.write('{}'.format(nomecarga[i]))
    arquivo.write(" bus1=")
    arquivo.write(barracarga[i])
    arquivo.write(" kv=")
    arquivo.write('{:.6}'.format(GB[i][1]))
    arquivo.write(" kw=")
    arquivo.write('{:.6}'.format(carga[i]))
    arquivo.write(" kvar=")
    arquivo.write('{:.6}'.format(cargar[i]))
    arquivo.write(" pf=")
    arquivo.write('{:.6}'.format(FPcar[i]))
    arquivo.write(" phases=3 \n\n")
    i=i+1

i=0
if ger>0:
    arquivo.write("// ---------- Gerador ---------- \n\n")
while (i<ger):
    arquivo.write("New generator.G")
    arquivo.write('{}'.format(nomeger[i]))
    arquivo.write(" bus1=")
    arquivo.write(barrager[i])
    arquivo.write(" kv=")
    arquivo.write(GB[i][1])
    arquivo.write(" kw=")
    arquivo.write('{:.7}'.format(Pger[i]))
    arquivo.write(" kvar=")
    arquivo.write('{:.6}'.format(Qger[i]))
    arquivo.write(" model=1 phases=3 \n\n")
```

```
    i=i+1

i=0
if caps>0:
    arquivo.write("// ---------- Capacitor ---------- \n\n")
while (i<cap):
    arquivo.write("New capacitor.CAP")
    arquivo.write('{}'.format(i))
    arquivo.write(" bus1=")
    arquivo.write(barradecap[i])
    arquivo.write(" kv=")
    arquivo.write('{:.6}'.format(GBcap[i]))
    arquivo.write(" kvar=")
    arquivo.write('{:.6}'.format(mvar[i]))
    arquivo.write(' phases=3')
    arquivo.write(" \n\n")
    i=i+1

arquivo.write("// ---------- Tipo de Solução ---------- \n\n")
arquivo.write("Set mode=snapshot \n\n")

arquivo.write("// ---------- Valores bases de tensão ---------- \n\n")
i=0
arquivo.write("Set voltagebases=[")
while (i<cvbn):
    arquivo.write(' {}'.format(cvb[i]))
    i=i+1
arquivo.write("] \n")
arquivo.write("calcvoltagebases \n\n")

arquivo.write("// ---------- Solucionar ---------- \n\n")
arquivo.write("Solve\n")

arquivo.close()
```

ANEXO B – RESULTADOS

B.1 – SISTEMA 5 BARRAS

B.1.1 – Vetor A

['TITU' 'Caso Sistema Teste 5 Barras' 'DBAR' '(Num)OETGb(nome)Gl(V)(A)(Pg)(Qg)(Qn)(Qm)(Bc)(Pl)(Ql)(Sh)Are(Vf)' ' 1 L2 Maq1 1017 0. 30.6.522-99999999. 30 11000' ' 2 L1 Maq2 10253.66 90.19.56-99999999. 30 11000' ' 10 L Barra_Term_1 1006-3.4 11000' ' 20 L Barra_Term_2 1014 .18 11000' ' 30 L Barra_PQ 1000-9.5 120. 0. 11000' '99999' 'DLIN' '(De)d O d(Pa)NcEP (R%)(X%)(Mvar)(Tap)(Tmn)(Tmx)(Phs)(Bc)(Cn)(Ce)Ns' ' 1 10 1 .1 20.' ' 2 20 1 .1 7.' ' 10 20 1 .1 14. 1.02' ' 10 20 2 .1 14. .9' ' 10 30 1 .1 18.' ' 10 30 2 .1 18.' '99999' 'FIM']

B.1.2 – Matriz E

[['T', 'I', 'T', 'U'],

['C', 'a', 's', 'o', ' ', 'S', 'i', 's', 't', 'e', 'm', 'a', ' ', 'T', 'e', 's', 't', 'e', ' ', '5', ' ', 'B', 'a', 'r', 'r', 'a', 's'],

['D', 'B', 'A', 'R'],

['(', 'N', 'u', 'm', ')', 'O', 'E', 'T', 'G', 'b', '(', ' ', ' ', ' ', 'n', 'o', 'm', 'e', ' ', ' ', ' ', ')', 'G', 'l', '(', ' ', 'V', ')', '(', ' ', 'A', ')', '(', ' ', 'P', 'g', ')', '(', ' ', 'Q', 'g', ')', '(', ' ', 'Q', 'n', ')', '(', ' ', 'Q', 'm', ')', '(', 'B', 'c', ' ', ' ', ')', '(', ' ', 'P', 'l', ')', '(', ' ', 'Q', 'l', ')', '(', ' ', 'S', 'h', ')', 'A', 'r', 'e', '(', 'V', 'f', ')'],

[' ', ' ', ' ', ' ', '1', ' ', 'L', '2', ' ', ' ', 'M', 'a', 'q', '1', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '0', '1', '7', ' ', ' ', '0', '.', ' ', ' ', '3', '0', '.', '6', '.', '5', '2', '2', '-', '9', '9', '9', '9', '9', '9', '9', '9', '.', ' ', ' ', ' ', ' ', '3', '0', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0', '0', '0'],

[' ', ' ', ' ', ' ', '2', ' ', 'L', '1', ' ', ' ', 'M', 'a', 'q', '2', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '0', '2', '5', '3', '.', '6', '6', ' ', ' ', '9', '0', '.', '1', '9', '.', '5', '6', '-', '9', '9', '9', '9', '9', '9', '9', '9', '.', ' ', ' ', ' ', '3', '0', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0', '0', '0'],

[' ', ' ', ' ', '1', '0', ' ', 'L', ' ', ' ', ' ', 'B', 'a', 'r', 'r', 'a', '_', 'T', 'e', 'r', 'm', '_', '1', ' ', ' ', '1', '0', '0', '6', '-', '3', '.', '4', ' ', '1', '1', '0', '0', '0'],

[' ', ' ', ' ', '2', '0', ' ', 'L', ' ', ' ', ' ', 'B', 'a', 'r', 'r', 'a', '_', 'T', 'e', 'r', 'm', '_', '2', ' ', ' ', '1', '0', '1', '4', ' ', '.', '1', '8', ' ', '1', '1', '0', '0', '0'],

[' ', ' ', ' ', '3', '0', ' ', 'L', ' ', ' ', ' ', 'B', 'a', 'r', 'r', 'a', '_', 'P', 'Q', ' ', ' ', ' ', ' ', ' ', ' ', '1', '0', '0', '0', '-', '9', '.', '5', ' ', '1', '2', '0', '.', ' ', ' ', ' ', '0', '.', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0', '0', '0'],
['9', '9', '9', '9', '9'],
['D', 'L', 'I', 'N'],
['(', 'D', 'e', ' ', ')', 'd', ' ', 'O', ' ', 'd', '(', 'P', 'a', ' ', ')', 'N', 'c', 'E', 'P', ' ', '(', ' ', 'R', '%', ' ', ')', '(', ' ', 'X', '%', ' ', ')', '(', 'M', 'v', 'a', 'r', ')', '(', 'T', 'a', 'p', ')', '(', 'T', 'm', 'n', ')', '(', 'T', 'm', 'x', ')', '(', 'P', 'h', 's', ')', '(', 'B', 'c', ' ', ' ', ')', '(', 'C', 'n', ')', '(', 'C', 'e', ')', 'N', 's'],
[' ', ' ', ' ', ' ', '1', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '0', ' ', '1', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '.', '1', ' ', ' ', ' ', '2', '0', '.'],
[' ', ' ', ' ', ' ', '2', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '2', '0', ' ', '1', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '.', '1', ' ', ' ', ' ', ' ', '7', '.'],
[' ', ' ', ' ', '1', '0', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '2', '0', ' ', '1', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '.', '1', ' ', '
', ' ', '1', '4', '.', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '.', '0', '2'],
[' ', ' ', ' ', '1', '0', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '2', '0', ' ', '2', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '.', '1', ' ', '
', ' ', '1', '4', '.', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '.', '9'],
[' ', ' ', ' ', '1', '0', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '3', '0', ' ', '1', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '.', '1', ' ', '
', ' ', '1', '8', '.'],
[' ', ' ', ' ', '1', '0', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '3', '0', ' ', '2', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '.', '1', ' ', '
', ' ', '1', '8', '.'],
['9', '9', '9', '9', '9'],
['F', 'I', 'M']]

B.1.3 – DBAR

[['1', '0', 'L', '2', '0', 'Maq1', '0', '1017', '0.', '30.', '6.522', '-9999', '9999.', '30', '0.', '0.', '0.', '1', '1000'],
['2', '0', 'L', '1', '0', 'Maq2', '0', '1025', '3.66', '90.', '19.56', '-9999', '9999.', '30', '0.', '0.', '0.', '1', '1000'],
['10', '0', 'L', '0', '0', 'Barra_Term_1', '0', '1006', '-3.4', '0.', '0.', '0.', '0.', '0.', '0.', '0.', '0.', '1', '1000'],
['20', '0', 'L', '0', '0', 'Barra_Term_2', '0', '1014', '.18', '0.', '0.', '0.', '0.', '0.', '0.', '0.', '0.', '1', '1000'],
['30', '0', 'L', '0', '0', 'Barra_PQ', '0', '1000', '-9.5', '0.', '0.', '0.', '0.', '0.', '120.', '0.', '0.', '1', '1000']]

B.1.4 – DLIN

[['1', 'L', 'A', 'L', '10', '1', 'L', 'F', '.1', '20.', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0'],
['2', 'L', 'A', 'L', '20', '1', 'L', 'F', '.1', '7.', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0'],
['10', 'L', 'A', 'L', '20', '1', 'L', 'F', '.1', '14.', '0', '1.02', '0', '0', '0', '0', '0', '0', '0', '0', '0'],
['10', 'L', 'A', 'L', '20', '2', 'L', 'F', '.1', '14.', '0', '.9', '0', '0', '0', '0', '0', '0', '0', '0', '0'],
['10', 'L', 'A', 'L', '30', '1', 'L', 'F', '.1', '18.', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0'],
['10', 'L', 'A', 'L', '30', '2', 'L', 'F', '.1', '18.', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0']]

B.1.5 – Potências dos equipamentos

CIRCUIT ELEMENT POWER FLOW
(Power Flow into element from indicated Bus)
Power Delivery Elements
Bus Phase kW +j kvar kVA PF
ELEMENT = "Vsource.SOURCE"1 1 -10024.9 +j -7935.0 12785.2 0.7841 1 2 -10024.9 +j -7935.0 12785.2 0.7841 1 3 -10024.9 +j -7935.0 12785.2 0.7841 TERMINAL TOTAL-30074.6 +j -23805.0 38355.7 0.7841 1 0 0.0 +j 0.0 0.0 1.0000 1 0 0.0 +j 0.0 0.0 1.0000 1 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 0.0 +j 0.0 0.0 1.0000
ELEMENT = "Transformer.TR0"10 1 -14422.6 +j -13771.8 19941.8 0.7232 10 2 -14422.6 +j -13771.8 19941.8 0.7232 10 3 -14422.6 +j -13771.8 19941.8 0.7232 10 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL-43267.8 +j -41315.4 59825.3 0.7232 20 1 14477.0 +j 15673.9 21336.7 0.6785 20 2 14477.0 +j 15673.9 21336.7 0.6785 20 3 14477.0 +j 15673.9 21336.7 0.6785 20 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 43430.9 +j 47021.8 64010.0 0.6785
ELEMENT = "Transformer.TR1"10 1 -15443.7 +j 15959.3 22208.2 -0.6954 10 2 -15443.7 +j 15959.3 22208.2 -0.6954 10 3 -15443.7 +j 15959.3 22208.2 -0.6954 10 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL-46331.1 +j 47877.8 66624.7 -0.6954 20 1 15496.2 +j -14122.6 20966.2 -0.7391 20 2 15496.2 +j -14122.6 20966.2 -0.7391 20 3 15496.2 +j -14122.6 20966.2 -0.7391 20 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 46488.5 +j -42367.8 62898.5 -0.7391
ELEMENT = "Line.L0"1 1 10024.9 +j 7935.0 12785.2 0.7841 1 2 10024.9 +j 7935.0 12785.2 0.7841 1 3 10024.9 +j 7935.0 12785.2 0.7841 TERMINAL TOTAL 30074.6 +j 23805.0 38355.7 0.7841 10 1 -10020.0 +j -6957.7 12198.8 0.8214 10 2 -10020.0 +j -6957.7 12198.8 0.8214 10 3 -10020.0 +j -6957.7 12198.8 0.8214 TERMINAL TOTAL-30060.0 +j -20873.0 36596.3 0.8214
ELEMENT = "Line.L1"2 1 30000.0 +j 3433.1 30195.8 0.9935 2 2 30000.0 +j 3433.1 30195.8 0.9935 2 3 30000.0 +j 3433.1 30195.8 0.9935 TERMINAL TOTAL 90000.0 +j 10299.3 90587.4 0.9935 20 1 -29973.1 +j -1551.3 30013.2 0.9987 20 2 -29973.1 +j -1551.3

30013.2 0.9987 20 3 -29973.1 +j -1551.3 30013.2 0.9987 TERMINAL TOTAL-89919.4 +j -4653.9 90039.7 0.9987

ELEMENT = "Line.L2"10 1 19943.1 +j 2385.1 20085.3 0.9929 10 2 19943.1 +j 2385.1 20085.3 0.9929 10 3 19943.1 +j 2385.1 20085.3 0.9929 TERMINAL TOTAL 59829.4 +j 7155.3 60255.8 0.9929 30 1 -19929.9 +j -0.6 19929.9 1.0000 30 2 -19929.9 +j -0.6 19929.9 1.0000 30 3 -19929.9 +j -0.6 19929.9 1.0000 TERMINAL TOTAL-59789.7 +j -1.7 59789.7 1.0000

ELEMENT = "Line.L3"10 1 19943.1 +j 2385.1 20085.3 0.9929 10 2 19943.1 +j 2385.1 20085.3 0.9929 10 3 19943.1 +j 2385.1 20085.3 0.9929 TERMINAL TOTAL 59829.4 +j 7155.3 60255.8 0.9929 30 1 -19929.9 +j -0.6 19929.9 1.0000 30 2 -19929.9 +j -0.6 19929.9 1.0000 30 3 -19929.9 +j -0.6 19929.9 1.0000 TERMINAL TOTAL-59789.7 +j -1.7 59789.7 1.0000

= =

Power Conversion Elements

Bus Phase kW +j kvar kVA PF

ELEMENT = "Load.C4"30 1 39860.1 +j 11.8 39860.1 1.0000 30 2 39860.1 +j 11.8 39860.1 1.0000 30 3 39860.1 +j 11.8 39860.1 1.0000 30 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 119580.2 +j 35.5 119580.3 1.0000

ELEMENT = "Generator.G1"2 1 -29998.6 +j -3445.7 30195.8 0.9935 2 2 -29998.6 +j -3445.7 30195.8 0.9935 2 3 -29998.6 +j -3445.7 30195.8 0.9935 2 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL -89995.7 +j -10337.0 90587.4 0.9935

Total Circuit Losses = 495.3 +j 34100.9

B.1.6 – Tensões de fase e pu dos equipamentos

NODE-GROUND VOLTAGES BY CIRCUIT ELEMENT

Power Delivery Elements

Bus (node ref) Phase Magnitude, kV (pu) Angle

ELEMENT = "Vsource.SOURCE"1 (1) 1 0.57837 (1.002) /_ -0.7 1 (2) 2 0.57837 (1.002) /_ -120.7 1 (3) 3 0.57837 (1.002) /_ 119.3 ———— 1 (0) 0 0 (0) /_ 0.0 1 (0) 0 0 (0) /_ 0.0 1 (0) 0 0 (0) /_ 0.0

ELEMENT = "Transformer.TR0"10 (4) 1 0.55184 (0.9558) /_ -4.2 10 (5) 2 0.55184 (0.9558) /_ -124.2 10 (6) 3 0.55184 (0.9558) /_ 115.8 10 (0) 0 0 (0) /_ 0.0 ———— 20 (7) 1 0.57886 (1.003) /_ -0.6 20 (8) 2 0.57886 (1.003) /_ -120.6 20 (9) 3 0.57886 (1.003) /_ 119.4 20 (0) 0 0 (0) /_ 0.0

ELEMENT = "Transformer.TR1"10 (4) 1 0.55184 (0.9558) /_ -4.2 10 (5) 2 0.55184 (0.9558) /_ -124.2 10 (6) 3 0.55184 (0.9558) /_ 115.8 10 (0) 0 0 (0) /_ 0.0 ———— 20 (7) 1 0.57886 (1.003) /_ -0.6 20 (8) 2 0.57886 (1.003) /_ -120.6 20 (9) 3 0.57886 (1.003)

/_ 119.4 20 (0) 0 0 (0) /_ 0.0

ELEMENT = "Line.L0"1 (1) 1 0.57837 (1.002) /_ -0.7 1 (2) 2 0.57837 (1.002) /_ -120.7 1 (3) 3 0.57837 (1.002) /_ 119.3 ———— 10 (4) 1 0.55184 (0.9558) /_ -4.2 10 (5) 2 0.55184 (0.9558) /_ -124.2 10 (6) 3 0.55184 (0.9558) /_ 115.8

ELEMENT = "Line.L1"2 (10) 1 0.58238 (1.009) /_ 2.9 2 (11) 2 0.58238 (1.009) /_ -117.1 2 (12) 3 0.58238 (1.009) /_ 122.9 ———— 20 (7) 1 0.57886 (1.003) /_ -0.6 20 (8) 2 0.57886 (1.003) /_ -120.6 20 (9) 3 0.57886 (1.003) /_ 119.4

ELEMENT = "Line.L2"10 (4) 1 0.55184 (0.9558) /_ -4.2 10 (5) 2 0.55184 (0.9558) /_ -124.2 10 (6) 3 0.55184 (0.9558) /_ 115.8 ———— 30 (13) 1 0.54757 (0.9484) /_ -11.1 30 (14) 2 0.54757 (0.9484) /_ -131.1 30 (15) 3 0.54757 (0.9484) /_ 108.9

ELEMENT = "Line.L3"10 (4) 1 0.55184 (0.9558) /_ -4.2 10 (5) 2 0.55184 (0.9558) /_ -124.2 10 (6) 3 0.55184 (0.9558) /_ 115.8 ———— 30 (13) 1 0.54757 (0.9484) /_ -11.1 30 (14) 2 0.54757 (0.9484) /_ -131.1 30 (15) 3 0.54757 (0.9484) /_ 108.9

= =

Power Conversion Elements

Bus (node ref) Phase Magnitude, kV (pu) Angle

ELEMENT = "Load.C4"30 (13) 1 0.54757 (0.9484) /_ -11.1 30 (14) 2 0.54757 (0.9484) /_ -131.1 30 (15) 3 0.54757 (0.9484) /_ 108.9 30 (0) 0 0 (0) /_ 0.0

ELEMENT = "Generator.G1"2 (10) 1 0.58238 (1.009) /_ 2.9 2 (11) 2 0.58238 (1.009) /_ -117.1 2 (12) 3 0.58238 (1.009) /_ 122.9 2 (0) 0 0 (0) /_ 0.0

B.2 – SISTEMA DE 9 BARRAS

B.2.1 – Vetor A

[’TITU’ ’Sistema 9 Barras com gerador swing adicional’ ’DCTE’ ’(Mn) (Val) (Mn) (Val) (Mn) (Val) (Mn) (Val) (Mn) (Val)’ ’BASE 100. DASE 100. TEPA 1e-6 EXST .4 TETP 5. TBPA 5.’ ’TLPP 1. TEPR 1e-6 QLST .4 TLPR 1. TLPQ 2. TSBZ .01’ ’TSBA 5. ASTP .05 VSTP 5. TLVC .5 TLTC .01 TSFR .1E-7’ ’ZMAX 500. TLPV .5 VDVM 200. VDVN 40. TUDC .001 TADC .01’ ’PGER 30. TPST .2 VFLD 70. ZMIN .001 HIST 470 LFIT 10’ ’ACIT 30 LFCV 1 DCIT 10 VSIT 10 LPIT 50 LFLP 10’ ’PDIT 10 LCRT 96 LPRT 60 CSTP 500.’ ’ICIT 9000 DMAX 5 FDIV 2. ICMN 1e-5 VART 5. TSTP 33’ ’ICMV .5 APAS 90. CPAR 70. VAVT 2. VAVF 5. VMVF 15.’ ’VPVT 2. VPVF 5. VPMF 10. VSVF 20. VINF 1. VSUP 1.’ ’NDIR 8 STTR 1. TRPT 100. STIR 10.’ ’99999’ ’DBAR’ ’(Num)OETGb(nome)Gl(V)(A)(Pg)(Qg)(Qn)(Qm)(Bc)(Pl)(Ql)(Sh)Are(Vf)’ ’ 1 L2 Barra 1 11075 0.142.510.88-130.130.4 11000’ ’ 2 L1 Barra 2 11075-1.8 90.-2.59-101.101.2 7 21000’ ’ 3 L1 Barra 3 11075-1.4 85.-13.7-67.4 67.4 9 31000’ ’ 4 L0 Barra 4 11072-4.1

41000’ ’ 5 L0 Barra 5 11050-7.7 125. 50. 41000’ ’ 6 L0 Barra 6 11065-6.7 90. 30. 41000’ ’ 7 L0 Barra 7 11078-4.6 41000’ ’ 8 L0 Barra 8 11069-6.3 100. 35. 41000’ ’ 9 L0 Barra 9 11083-3.9 41000’ ’ 10 L2 Barra 10 11075 0.0.0000.000-999999999 51000’ ’99999’ ’DLIN’ ’(De)d O d(Pa)NcEP (R%)(X%)(Mvar)(Tap)(Tmn)(Tmx)(Phs)(Bc)(Cn)(Ce)Ns’ ’ 1 4 1 T 0. 5.76 1.0 250 250’ ’ 10 1 1 T 0. 0.01 999 999’ ’ 2 7 1 T 0. 6.25 1.0 200 200’ ’ 3 9 1 T 0. 5.86 1.0 300 300’ ’ 4 5 1 T 1. 8.5 17.6 300 300’ ’ 4 6 1 T 1.7 9.2 15.8 200 200’ ’ 6 9 1 T 3.9 17. 35.8 200 200’ ’ 7 5 1 T 3.2 16.1 30.6 300 300’ ’ 7 8 1 T .85 7.2 14.9 300 300’ ’ 8 9 1 T 1.19 10.08 20.9 300 300’ ’99999’ ’DARE’ ’(chg) (Identificacao da area) (Xmin) (Xmax)’ ’ 1 0. Area 1’ ’ 2 0. Area 2’ ’ 3 0. Area 3’ ’ 4 0. Area 4’ ’(5 0. Area 5’ ’99999’ ’(’ ’(==========’ ’(LISTA DE CONTINGÃNCIAS’ ’(==========’ ’DCTG IMPR’ ’(Nc) O Pr (IDENTIFICACAO DA CONTINGENCIA)’ ’1 1 LT_4_5_1’ ’(Tp) (El) (Pa) Nc (Ext) (DV1) (DV2) (DV3) (DV4) (DV5) (DV6) (DV7)’ ’CIRC 4 5 1’ ’FCAS’ ’(Nc) O Pr (IDENTIFICACAO DA CONTINGENCIA)’ ’2 1 LT_4_6_1’ ’(Tp) (El) (Pa) Nc (Ext) (DV1) (DV2) (DV3) (DV4) (DV5) (DV6) (DV7)’ ’CIRC 4 6 1’ ’FCAS’ ’(Nc) O Pr (IDENTIFICACAO DA CONTINGENCIA)’ ’3 1 LT_6_9_1’ ’(Tp) (El) (Pa) Nc (Ext) (DV1) (DV2) (DV3) (DV4) (DV5) (DV6) (DV7)’ ’CIRC 6 9 1’ ’FCAS’ ’(Nc) O Pr (IDENTIFICACAO DA CONTINGENCIA)’ ’4 1 LT_7_5_1’ ’(Tp) (El) (Pa) Nc (Ext) (DV1) (DV2) (DV3) (DV4) (DV5) (DV6) (DV7)’ ’CIRC 7 5 1’ ’FCAS’ ’(Nc) O Pr (IDENTIFICACAO DA CONTINGENCIA)’ ’5 1 LT_7_8_1’ ’(Tp) (El) (Pa) Nc (Ext) (DV1) (DV2) (DV3) (DV4) (DV5) (DV6) (DV7)’ ’CIRC 7 8 1’ ’FCAS’ ’(Nc) O Pr (IDENTIFICACAO DA CONTINGENCIA)’ ’6 1 LT_8_9_1’ ’(Tp) (El) (Pa) Nc (Ext) (DV1) (DV2) (DV3) (DV4) (DV5) (DV6) (DV7)’ ’CIRC 8 9 1’ ’FCAS’ ’99999’ ’(’ ’(==========’ ’(DADOS DE GRUPO LIMITE DE TENSAO’ ’(==========’ ’(’ ’DGLT’ ’(G (Vmn) (Vmx) (Vmne (Vmxe’ ’ 1 .9 1.1 .9 1.1’ ’99999’ ’(’ ’DGER’ ’(No) O (Pmn) (Pmx) (Fp) (FpR) (FPn) (Fa) (Fr) (Ag) (Xq) (Sno)’ ’ 1 0. 210.4’ ’ 2 0. 163.2’ ’ 3 0. 108.8’ ’99999’ ’EXLF NEWT CREM CTAP QLIM’ ’DVSA’ ’(Rg) (tp) (no) C (tp) (no) C (tp) (no) C (tp) (no)’ ’GUG1 BARR 1’ ’GUG2 BARR 2’ ’GUG3 BARR 3’ ’99999’ ’EXRS QLIM jump’ ’FIM’]

B.2.2 – Matriz E

['T', 'I', 'T', 'U'],

['S', 'i', 's', 't', 'e', 'm', 'a', ' ', '9', ' ', 'B', 'a', 'r', 'r', 'a', 's', ' ', 'c', 'o', 'm', ' ', 'g', 'e', 'r', 'a', 'd', 'o', 'r', ' ', 's', 'w', 'i', 'n', 'g', ' ', 'a', 'd', 'i', 'c', 'i', 'o', 'n', 'a', 'l'],

['D', 'C', 'T', 'E'],

['(', 'M', 'n', ')', ' ', '(', ' ', 'V', 'a', 'l', ')', ' ', '(', 'M', 'n', ')', ' ', '(', ' ', 'V', 'a', 'l', ')', ' ', '(', 'M', 'n', ')', ' ', '(', ' ', 'V', 'a', 'l', ')', ' ', '(', 'M', 'n', ')', ' ', '(', ' ', 'V', 'a', 'l', ')', ' ', '(', 'M', 'n', ')', ' ', '(', '

', 'V', 'a', 'l', ')', ' ', '(', 'M', 'n', ')', ' ', '(', ' ', 'V', 'a', 'l', ')'],

['B', 'A', 'S', 'E', ' ', ' ', ' ', '1', '0', '0', '.', ' ', 'D', 'A', 'S', 'E', ' ', ' ', ' ', '1', '0', '0', '.', ' ', 'T', 'E', 'P', 'A', ' ', ' ', ' ', '1', 'e', '-', '6', ' ', 'E', 'X', 'S', 'T', ' ', ' ', ' ', ' ', ' ', '.', '4', ' ', 'T', 'E', 'T', 'P', ' ', ' ', ' ', ' ', ' ', '5', '.', ' ', 'T', 'B', 'P', 'A', ' ', ' ', ' ', ' ', ' ', ' ', '5', '.'],

['T', 'L', 'P', 'P', ' ', ' ', ' ', ' ', ' ', '1', '.', ' ', 'T', 'E', 'P', 'R', ' ', ' ', ' ', '1', 'e', '-', '6', ' ', 'Q', 'L', 'S', 'T', ' ', ' ', ' ', ' ', ' ', '.', '4', ' ', 'T', 'L', 'P', 'R', ' ', ' ', ' ', ' ', ' ', '1', '.', ' ', 'T', 'L', 'P', 'Q', ' ', ' ', ' ', ' ', ' ', '2', '.', ' ', 'T', 'S', 'B', 'Z', ' ', ' ', ' ', ' ', '.', '0', '1'],

['T', 'S', 'B', 'A', ' ', ' ', ' ', ' ', '5', '.', ' ', 'A', 'S', 'T', 'P', ' ', ' ', ' ', ' ', '.', '0', '5', ' ', 'V', 'S', 'T', 'P', ' ', ' ', ' ', ' ', ' ', '5', '.', ' ', 'T', 'L', 'V', 'C', ' ', ' ', ' ', ' ', ' ', '.', '5', ' ', 'T', 'L', 'T', 'C', ' ', ' ', ' ', ' ', '.', '0', '1', ' ', 'T', 'S', 'F', 'R', ' ', ' ', '.', '1', 'E', '-', '7'],

['Z', 'M', 'A', 'X', ' ', ' ', ' ', '5', '0', '0', '.', ' ', 'T', 'L', 'P', 'V', ' ', ' ', ' ', ' ', ' ', '.', '5', ' ', 'V', 'D', 'V', 'M', ' ', ' ', ' ', '2', '0', '0', '.', ' ', 'V', 'D', 'V', 'N', ' ', ' ', ' ', ' ', '4', '0', '.', ' ', 'T', 'U', 'D', 'C', ' ', ' ', ' ', '.', '0', '0', '1', ' ', 'T', 'A', 'D', 'C', ' ', ' ', ' ', ' ', '.', '0', '1'],

['P', 'G', 'E', 'R', ' ', ' ', ' ', ' ', '3', '0', '.', ' ', 'T', 'P', 'S', 'T', ' ', ' ', ' ', ' ', ' ', '.', '2', ' ', 'V', 'F', 'L', 'D', ' ', ' ', ' ', ' ', '7', '0', '.', ' ', 'Z', 'M', 'I', 'N', ' ', ' ', ' ', '.', '0', '0', '1', ' ', 'H', 'I', 'S', 'T', ' ', ' ', ' ', ' ', '4', '7', '0', ' ', 'L', 'F', 'I', 'T', ' ', ' ', ' ', ' ', ' ', '1', '0'],

['A', 'C', 'I', 'T', ' ', ' ', ' ', ' ', ' ', '3', '0', ' ', 'L', 'F', 'C', 'V', ' ', ' ', ' ', ' ', ' ', ' ', '1', ' ', 'D', 'C', 'I', 'T', ' ', ' ', ' ', ' ', ' ', '1', '0', ' ', 'V', 'S', 'I', 'T', ' ', ' ', ' ', ' ', ' ', '1', '0', ' ', 'L', 'P', 'I', 'T', ' ', ' ', ' ', ' ', ' ', '5', '0', ' ', 'L', 'F', 'L', 'P', ' ', ' ', ' ', ' ', ' ', '1', '0'],

['P', 'D', 'I', 'T', ' ', ' ', ' ', ' ', '1', '0', ' ', 'L', 'C', 'R', 'T', ' ', ' ', ' ', ' ', ' ', '9', '6', ' ', 'L', 'P', 'R', 'T', ' ', ' ', ' ', ' ', ' ', '6', '0', ' ', 'C', 'S', 'T', 'P', ' ', ' ', ' ', '5', '0', '0', '.'],

['I', 'C', 'I', 'T', ' ', ' ', ' ', '9', '0', '0', '0', ' ', 'D', 'M', 'A', 'X', ' ', ' ', ' ', ' ', ' ', '5', ' ', 'F', 'D', 'I', 'V', ' ', ' ', ' ', ' ', ' ', '2', '.', ' ', 'I', 'C', 'M', 'N', ' ', ' ', ' ', '1', 'e', '-', '5', ' ', 'V', 'A', 'R', 'T', ' ', ' ', ' ', ' ', ' ', '5', '.', ' ', 'T', 'S', 'T', 'P', ' ', ' ', ' ', ' ', ' ', '3', '3'],

['I', 'C', 'M', 'V', ' ', ' ', ' ', ' ', ' ', '.', '5', ' ', 'A', 'P', 'A', 'S', ' ', ' ', ' ', ' ', '9', '0', '.', ' ', 'C', 'P', 'A', 'R', ' ', ' ', ' ', ' ', '7', '0', '.', ' ', 'V', 'A', 'V', 'T', ' ', ' ', ' ', ' ', ' ', '2', '.', ' ', 'V', 'A', 'V', 'F', ' ', ' ', ' ', ' ', ' ', '5', '.', ' ', 'V', 'M', 'V', 'F', ' ', ' ', ' ', ' ', '1', '5', '.'],

['V', 'P', 'V', 'T', ' ', ' ', ' ', ' ', ' ', '2', '.', ' ', 'V', 'P', 'V', 'F', ' ', ' ', ' ', ' ', ' ', '5', '.', ' ', 'V', 'P', 'M', 'F', ' ', ' ', ' ', ' ', '1', '0', '.', ' ', 'V', 'S', 'V', 'F', ' ', ' ', ' ', ' ', '2', '0', '.', ' ', 'V', 'I', 'N', 'F', ' ', ' ', ' ', ' ', ' ', '1', '.', ' ', 'V', 'S', 'U', 'P', ' ', ' ', ' ', ' ', ' ', '1', '.'],

['N', 'D', 'I', 'R', ' ', ' ', ' ', ' ', ' ', '8', ' ', ' ', 'S', 'T', 'T', 'R', ' ', ' ', ' ', ' ', ' ', '1', '.', ' ', 'T', 'R', 'P', 'T', ' ', ' ', ' ', '1', '0', '0', '.', ' ', 'S', 'T', 'I', 'R', ' ', ' ', ' ', ' ', '1', '0', '.'],

['9', '9', '9', '9', '9'],

['D', 'B', 'A', 'R'],

['(', 'N', 'u', 'm', ')', 'O', 'E', 'T', 'G', 'b', '(', ' ', ' ', ' ', ' ', 'n', 'o', 'm', 'e', ' ', ' ', ' ', ' ', ')', 'G', 'I', '(', ' ', 'V', ')', '(', ' ', 'A', ')', '(', ' ', 'P', 'g', ')', '(', ' ', 'Q', 'g', ')', '(', ' ', 'Q', 'n', ')', '(', ' ', 'Q', 'm', ')', '(', 'B', 'c', ' ', ' ', ' ', ')', '(', ' ', 'P', 'I', ')', '(', ' ', 'Q', 'I', ')', '(', ' ', 'S', 'h', ')', 'A', 'r', 'e', '(', 'V', 'f', ')'],

[' ', ' ', ' ', ' ', ' ', '1', ' ', ' ', 'L', '2', ' ', ' ', ' ', 'B', 'a', 'r', 'r', 'a', ' ', ' ', '1', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0', '7', '5', ' ', ' ', ' ', '0', '.', '1', '4', '2', '.', '5', '1', '0', '.', '8', '8', '-', '1', '3', '0', '.', '1', '3', '0', '.', '4', ' ', ' ',

' ', '1', '1', '0', '0', '0'],
[' ', ' ', ' ', ' ', '2', ' ', 'L', '1', ' ', ' ', 'B', 'a', 'r', 'r', 'a', ' ', '2', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0',
'7', '5', '-', '1', '.', '8', ' ', ' ', '9', '0', '.', '-', '2', '.', '5', '9', '-', '1', '0', '1', '.', '1', '0', '1', '.', '2', ' ', ' ', '
', '7', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '2', '1', '0', '0', '0'],
[' ', ' ', ' ', ' ', '3', ' ', 'L', '1', ' ', ' ', 'B', 'a', 'r', 'r', 'a', ' ', '3', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0',
'7', '5', '-', '1', '.', '4', ' ', ' ', '8', '5', '.', '-', '1', '3', '.', '7', '-', '6', '7', '.', '4', ' ', '6', '7', '.', '4', ' ', ' ', '
', '9', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '3', '1', '0', '0', '0'],
[' ', ' ', ' ', ' ', '4', ' ', 'L', '0', ' ', ' ', 'B', 'a', 'r', 'r', 'a', ' ', '4', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0',
'7', '2', '-', '4', '.', '1', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '4', '1', '0', '0', '0'],
[' ', ' ', ' ', ' ', '5', ' ', 'L', '0', ' ', ' ', 'B', 'a', 'r', 'r', 'a', ' ', '5', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0',
'5', '0', '-', '7', '.', '7', ' ',
' ', ' ', '1', '2', '5', '.', ' ', ' ', '5', '0', '.', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '4', '1', '0', '0', '0'],
[' ', ' ', ' ', ' ', '6', ' ', 'L', '0', ' ', ' ', 'B', 'a', 'r', 'r', 'a', ' ', '6', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0',
'6', '5', '-', '6', '.', '7', ' ',
' ', ' ', ' ', '9', '0', '.', ' ', ' ', '3', '0', '.', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '4', '1', '0', '0', '0'],
[' ', ' ', ' ', ' ', '7', ' ', 'L', '0', ' ', ' ', 'B', 'a', 'r', 'r', 'a', ' ', '7', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0',
'7', '8', '-', '4', '.', '6', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '4', '1', '0', '0', '0'],
[' ', ' ', ' ', ' ', '8', ' ', 'L', '0', ' ', ' ', 'B', 'a', 'r', 'r', 'a', ' ', '8', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0',
'6', '9', '-', '6', '.', '3', ' ',
' ', ' ', '1', '0', '0', '.', ' ', ' ', '3', '5', '.', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '4', '1', '0', '0', '0'],
[' ', ' ', ' ', ' ', '9', ' ', 'L', '0', ' ', ' ', 'B', 'a', 'r', 'r', 'a', ' ', '9', ' ', ' ', ' ', ' ', ' ', ' ', '1', '1', '0',
'8', '3', '-', '3', '.', '9', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '4', '1', '0', '0', '0'],
[' ', ' ', ' ', '1', '0', ' ', 'L', '2', ' ', ' ', 'B', 'a', 'r', 'r', 'a', ' ', '1', '0', ' ', ' ', ' ', ' ', ' ', '1', '1', '0',
'7', '5', ' ', ' ', '0', '.', '0', '.', '0', '0', '0', '0', '.', '0', '0', '0', '-', '9', '9', '9', '9', '9', '9', '9', '9', '9', ' ', '
', ' ', '5', '1', '0', '0', '0'],
['9', '9', '9', '9', '9'],
['D', 'L', 'I', 'N'],
['(', 'D', 'e', ' ', ')', 'd', ' ', 'O', ' ', 'd', '(', 'P', 'a', ' ', ')', 'N', 'c', 'E', 'P', ' ', '(', ' ', 'R', '%', ' ',
')', '(', ' ', 'X', '%', ' ', ')', '(', 'M', 'v', 'a', 'r', ')', '(', 'T', 'a', 'p', ')', '(', 'T', 'm', 'n', ')', '(', 'T', 'm', 'x',
')', '(', 'P', 'h', 's', ')', '(', 'B', 'c', ' ', ' ', ')', '(', 'C', 'n', ')', '(', 'C', 'e', ')', 'N', 's'],
[' ', ' ', ' ', ' ', '1', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '4', ' ', '1', ' ', 'T', ' ', ' ', ' ', ' ', ' ', '0', '.', ' ', ' ',
'5', '.', '7', '6', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '.', '0', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
', ' ', ' ', ' ', ' ', ' ', '2', '5', '0', ' ', '2', '5', '0'],
[' ', ' ', ' ', '1', '0', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', ' ', '1', ' ', 'T', ' ', ' ', ' ', ' ', ' ', '0', '.', ' ', '
', '0', '.', '0', '1', ' ', '
', ' ', ' ', ' ', ' ', ' ', '9', '9', '9', ' ', '9', '9', '9'],

[’ ’, ’ ’, ’ ’, ’ ’, ’2’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’7’, ’ ’, ’1’, ’ ’, ’T’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’0’, ’.’, ’ ’, ’ ’, ’6’, ’.’, ’2’, ’5’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’1’, ’.’, ’0’, ’ ’, ’2’, ’0’, ’0’, ’ ’, ’2’, ’0’, ’0’],

[’ ’, ’ ’, ’ ’, ’ ’, ’3’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’9’, ’ ’, ’1’, ’ ’, ’T’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’0’, ’.’, ’ ’, ’ ’, ’5’, ’.’, ’8’, ’6’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’1’, ’.’, ’0’, ’ ’, ’3’, ’0’, ’0’, ’ ’, ’3’, ’0’, ’0’],

[’ ’, ’ ’, ’ ’, ’ ’, ’4’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’5’, ’ ’, ’1’, ’ ’, ’T’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’1’, ’.’, ’ ’, ’ ’, ’ ’, ’8’, ’.’, ’5’, ’ ’, ’ ’, ’1’, ’7’, ’.’, ’6’, ’ ’, ’3’, ’0’, ’0’, ’ ’, ’3’, ’0’, ’0’],

[’ ’, ’ ’, ’ ’, ’ ’, ’4’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’6’, ’ ’, ’1’, ’ ’, ’T’, ’ ’, ’ ’, ’ ’, ’ ’, ’1’, ’.’, ’7’, ’ ’, ’ ’, ’9’, ’.’, ’2’, ’ ’, ’ ’, ’1’, ’5’, ’.’, ’8’, ’ ’, ’2’, ’0’, ’0’, ’ ’, ’2’, ’0’, ’0’],

[’ ’, ’ ’, ’ ’, ’ ’, ’6’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’9’, ’ ’, ’1’, ’ ’, ’T’, ’ ’, ’ ’, ’ ’, ’ ’, ’3’, ’.’, ’9’, ’ ’, ’ ’, ’1’, ’7’, ’.’, ’ ’, ’ ’, ’3’, ’5’, ’.’, ’8’, ’ ’, ’2’, ’0’, ’0’, ’ ’, ’2’, ’0’, ’0’],

[’ ’, ’ ’, ’ ’, ’ ’, ’7’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’5’, ’ ’, ’1’, ’ ’, ’T’, ’ ’, ’ ’, ’ ’, ’ ’, ’3’, ’.’, ’2’, ’ ’, ’1’, ’6’, ’.’, ’1’, ’ ’, ’ ’, ’3’, ’0’, ’.’, ’6’, ’ ’, ’3’, ’0’, ’0’, ’ ’, ’3’, ’0’, ’0’],

[’ ’, ’ ’, ’ ’, ’ ’, ’7’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’8’, ’ ’, ’1’, ’ ’, ’T’, ’ ’, ’ ’, ’ ’, ’ ’, ’.’, ’8’, ’5’, ’ ’, ’ ’, ’7’, ’.’, ’2’, ’ ’, ’ ’, ’1’, ’4’, ’.’, ’9’, ’ ’, ’3’, ’0’, ’0’, ’ ’, ’3’, ’0’, ’0’],

[’ ’, ’ ’, ’ ’, ’ ’, ’8’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’9’, ’ ’, ’1’, ’ ’, ’T’, ’ ’, ’ ’, ’ ’, ’1’, ’.’, ’1’, ’9’, ’ ’, ’1’, ’0’, ’.’, ’0’, ’8’, ’ ’, ’ ’, ’2’, ’0’, ’.’, ’9’, ’ ’, ’3’, ’0’, ’0’, ’ ’, ’3’, ’0’, ’0’],

[’9’, ’9’, ’9’, ’9’, ’9’],

[’D’, ’A’, ’R’, ’E’],

[’(’, ’c’, ’h’, ’g’, ’)’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’(’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’I’, ’d’, ’e’, ’n’, ’t’, ’i’, ’f’, ’i’, ’c’, ’a’, ’c’, ’a’, ’o’, ’ ’, ’d’, ’a’, ’ ’, ’a’, ’r’, ’e’, ’a’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’)’, ’ ’, ’(’, ’X’, ’m’, ’i’, ’n’, ’)’, ’ ’, ’(’, ’X’, ’m’, ’a’, ’x’, ’)’],

[’ ’, ’ ’, ’1’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’0’, ’.’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’A’, ’r’, ’e’, ’a’, ’ ’, ’1’],

[’ ’, ’ ’, ’2’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’0’, ’.’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’A’, ’r’, ’e’, ’a’, ’ ’, ’2’],

[’ ’, ’ ’, ’3’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’0’, ’.’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’A’, ’r’, ’e’, ’a’, ’ ’, ’3’],

[’ ’, ’ ’, ’4’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’0’, ’.’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’A’, ’r’, ’e’, ’a’, ’ ’, ’4’],

[’(’, ’ ’, ’ ’, ’5’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’0’, ’.’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’A’, ’r’, ’e’, ’a’, ’ ’, ’5’],

[’9’, ’9’, ’9’, ’9’, ’9’],

[’(’], [’(’, ’=’],

[’(’, ’ ’, ’L’, ’I’, ’S’, ’T’, ’A’, ’ ’, ’D’, ’E’, ’ ’, ’C’, ’O’, ’N’, ’T’, ’I’, ’N’, ’G’, ’Ã’, ’’, ’N’, ’C’, ’I’, ’A’,

’S’],

[’(’, ’=’],

[’D’, ’C’, ’T’, ’G’, ’ ’, ’I’, ’M’, ’P’, ’R’],

[’(’, ’N’, ’c’, ’)’, ’ ’, ’O’, ’ ’, ’P’, ’r’, ’ ’, ’(’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’I’, ’D’, ’E’, ’N’, ’T’, ’I’, ’F’, ’I’, ’C’, ’A’, ’C’, ’A’, ’O’, ’ ’, ’D’, ’A’, ’ ’, ’C’, ’O’, ’N’, ’T’, ’I’, ’N’, ’G’, ’E’, ’N’, ’C’, ’I’, ’A’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’)’],

[’1’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’1’, ’ ’, ’L’, ’T’, ’_’, ’4’, ’_’, ’5’, ’_’, ’1’],

[’(’, ’T’, ’p’, ’)’, ’ ’, ’(’, ’E’, ’I’, ’ ’, ’)’, ’ ’, ’(’, ’P’, ’a’, ’ ’, ’)’, ’ ’, ’N’, ’c’, ’ ’, ’(’, ’E’, ’x’, ’t’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’1’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’2’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’3’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’4’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’5’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’6’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’7’, ’)’],

[’C’, ’I’, ’R’, ’C’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’4’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’5’, ’ ’, ’ ’, ’1’],

[’F’, ’C’, ’A’, ’S’],

[’(’, ’N’, ’c’, ’)’, ’ ’, ’O’, ’ ’, ’P’, ’r’, ’ ’, ’(’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’I’, ’D’, ’E’, ’N’, ’T’, ’I’, ’F’, ’I’, ’C’, ’A’, ’C’, ’A’, ’O’, ’ ’, ’D’, ’A’, ’ ’, ’C’, ’O’, ’N’, ’T’, ’I’, ’N’, ’G’, ’E’, ’N’, ’C’, ’I’, ’A’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’)’],

[’2’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’1’, ’ ’, ’L’, ’T’, ’_’, ’4’, ’_’, ’6’, ’_’, ’1’],

[’(’, ’T’, ’p’, ’)’, ’ ’, ’(’, ’E’, ’I’, ’ ’, ’)’, ’ ’, ’(’, ’P’, ’a’, ’ ’, ’)’, ’ ’, ’N’, ’c’, ’ ’, ’(’, ’E’, ’x’, ’t’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’1’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’2’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’3’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’4’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’5’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’6’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’7’, ’)’],

[’C’, ’I’, ’R’, ’C’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’4’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’6’, ’ ’, ’ ’, ’1’],

[’F’, ’C’, ’A’, ’S’],

[’(’, ’N’, ’c’, ’)’, ’ ’, ’O’, ’ ’, ’P’, ’r’, ’ ’, ’(’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’I’, ’D’, ’E’, ’N’, ’T’, ’I’, ’F’, ’I’, ’C’, ’A’, ’C’, ’A’, ’O’, ’ ’, ’D’, ’A’, ’ ’, ’C’, ’O’, ’N’, ’T’, ’I’, ’N’, ’G’, ’E’, ’N’, ’C’, ’I’, ’A’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’)’],

[’3’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’1’, ’ ’, ’L’, ’T’, ’_’, ’6’, ’_’, ’9’, ’_’, ’1’],

[’(’, ’T’, ’p’, ’)’, ’ ’, ’(’, ’E’, ’I’, ’ ’, ’)’, ’ ’, ’(’, ’P’, ’a’, ’ ’, ’)’, ’ ’, ’N’, ’c’, ’ ’, ’(’, ’E’, ’x’, ’t’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’1’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’2’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’3’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’4’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’5’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’6’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’7’, ’)’],

[’C’, ’I’, ’R’, ’C’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’6’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’9’, ’ ’, ’ ’, ’1’],

[’F’, ’C’, ’A’, ’S’],

[’(’, ’N’, ’c’, ’)’, ’ ’, ’O’, ’ ’, ’P’, ’r’, ’ ’, ’(’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’I’, ’D’, ’E’, ’N’, ’T’, ’I’, ’F’, ’I’, ’C’, ’A’, ’C’, ’A’, ’O’, ’ ’, ’D’, ’A’, ’ ’, ’C’, ’O’, ’N’, ’T’, ’I’, ’N’, ’G’, ’E’, ’N’, ’C’, ’I’, ’A’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’)’],

[’4’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’1’, ’ ’, ’L’, ’T’, ’_’, ’7’, ’_’, ’5’, ’_’, ’1’],

[’(’, ’T’, ’p’, ’)’, ’ ’, ’(’, ’E’, ’I’, ’ ’, ’)’, ’ ’, ’(’, ’P’, ’a’, ’ ’, ’)’, ’ ’, ’N’, ’c’, ’ ’, ’(’, ’E’, ’x’, ’t’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’1’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’2’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’3’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’4’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’5’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’6’, ’)’, ’ ’, ’(’, ’D’, ’V’, ’7’, ’)’],

['C', 'I', 'R', 'C', ' ', ' ', ' ', ' ', ' ', '7', ' ', ' ', ' ', ' ', ' ', '5', ' ', ' ', '1'],

['F', 'C', 'A', 'S'],

['(', 'N', 'c', ')', ' ', 'O', ' ', 'P', 'r', ' ', '(', ' ', ' ', ' ', ' ', ' ', ' ', ' ', 'I', 'D', 'E', 'N', 'T', 'I', 'F', 'I', 'C', 'A', 'C', 'A', 'O', ' ', 'D', 'A', ' ', 'C', 'O', 'N', 'T', 'I', 'N', 'G', 'E', 'N', 'C', 'I', 'A', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ')'],

['5', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', ' ', 'L', 'T', '_', '7', '_', '8', '_', '1'],

['(', 'T', 'p', ')', ' ', '(', 'E', 'I', ' ', ')', ' ', '(', 'P', 'a', ' ', ')', ' ', 'N', 'c', ' ', '(', 'E', 'x', 't', ')', ' ', '(', 'D', 'V', '1', ')', ' ', '(', 'D', 'V', '2', ')', ' ', '(', 'D', 'V', '3', ')', ' ', '(', 'D', 'V', '4', ')', ' ', '(', 'D', 'V', '5', ')', ' ', '(', 'D', 'V', '6', ')', ' ', '(', 'D', 'V', '7', ')'],

['C', 'I', 'R', 'C', ' ', ' ', ' ', ' ', ' ', '7', ' ', ' ', ' ', ' ', ' ', '8', ' ', ' ', '1'],

['F', 'C', 'A', 'S'],

['(', 'N', 'c', ')', ' ', 'O', ' ', 'P', 'r', ' ', '(', ' ', ' ', ' ', ' ', ' ', ' ', ' ', 'I', 'D', 'E', 'N', 'T', 'I', 'F', 'I', 'C', 'A', 'C', 'A', 'O', ' ', 'D', 'A', ' ', 'C', 'O', 'N', 'T', 'I', 'N', 'G', 'E', 'N', 'C', 'I', 'A', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ')'],

['6', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', ' ', 'L', 'T', '_', '8', '_', '9', '_', '1'],

['(', 'T', 'p', ')', ' ', '(', 'E', 'I', ' ', ')', ' ', '(', 'P', 'a', ' ', ')', ' ', 'N', 'c', ' ', '(', 'E', 'x', 't', ')', ' ', '(', 'D', 'V', '1', ')', ' ', '(', 'D', 'V', '2', ')', ' ', '(', 'D', 'V', '3', ')', ' ', '(', 'D', 'V', '4', ')', ' ', '(', 'D', 'V', '5', ')', ' ', '(', 'D', 'V', '6', ')', ' ', '(', 'D', 'V', '7', ')'],

['C', 'I', 'R', 'C', ' ', ' ', ' ', ' ', ' ', '8', ' ', ' ', ' ', ' ', ' ', '9', ' ', ' ', '1'],

['F', 'C', 'A', 'S'],

['9', '9', '9', '9', '9'],

['('], ['(', '='],

['(', ' ', 'D', 'A', 'D', 'O', 'S', ' ', 'D', 'E', ' ', 'G', 'R', 'U', 'P', 'O', ' ', 'L', 'I', 'M', 'I', 'T', 'E', ' ', 'D', 'E', ' ', 'T', 'E', 'N', 'S', 'A', 'O'],

['(', '='], ['('],

['D', 'G', 'L', 'T'],

['(', 'G', ' ', '(', 'V', 'm', 'n', ')', ' ', '(', 'V', 'm', 'x', ')', ' ', '(', 'V', 'm', 'n', 'e', ' ', '(', 'V', 'm', 'x', 'e'],

[' ', '1', ' ', ' ', ' ', '.', '9', ' ', ' ', ' ', '1', '.', '1', ' ', ' ', ' ', '.', '9', ' ', ' ', ' ', ' ', '1', '.', '1'],

['9', '9', '9', '9', '9'],

['('], ['D', 'G', 'E', 'R'],

['(', 'N', 'o', ' ', ')', ' ', 'O', ' ', '(', 'P', 'm', 'n', ' ', ')', ' ', '(', 'P', 'm', 'x', ' ', ')', ' ', '(', ' ', 'F', 'p', ')', ' ', '(', 'F', 'p', 'R', ')', ' ', '(', 'F', 'P', 'n', ')', ' ', '(', 'F', 'a', ')', ' ', '(', 'F', 'r', ')', ' ', '(', 'A', 'g', ')', ' ', '(', ' ', 'X', 'q', ')', ' ', '(', 'S', 'n', 'o', ')'],

[' ', ' ', ' ', '1', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '0', '.', ' ', ' ', ' ', '2', '1', '0', '.', '4'],

[’ ’, ’ ’, ’ ’, ’2’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’0’, ’.’, ’ ’, ’ ’, ’ ’, ’1’, ’6’, ’3’, ’.’, ’2’],
[’ ’, ’ ’, ’ ’, ’3’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’ ’, ’0’, ’.’, ’ ’, ’ ’, ’ ’, ’1’, ’0’, ’8’, ’.’, ’8’],
[’9’, ’9’, ’9’, ’9’, ’9’],
[’E’, ’X’, ’L’, ’F’, ’ ’, ’N’, ’E’, ’W’, ’T’, ’ ’, ’C’, ’R’, ’E’, ’M’, ’ ’, ’C’, ’T’, ’A’, ’P’, ’ ’, ’Q’, ’L’, ’I’, ’M’],
[’D’, ’V’, ’S’, ’A’],
[’(’, ’R’, ’g’, ’)’, ’ ’, ’(’, ’t’, ’p’, ’)’, ’ ’, ’(’, ’n’, ’o’, ’ ’, ’)’, ’ ’, ’C’, ’ ’, ’(’, ’t’, ’p’, ’)’, ’ ’, ’(’, ’n’, ’o’, ’ ’, ’)’, ’ ’, ’C’, ’ ’, ’(’, ’t’, ’p’, ’)’, ’ ’, ’(’, ’n’, ’o’, ’ ’, ’)’, ’ ’, ’C’, ’ ’, ’(’, ’t’, ’p’, ’)’, ’ ’, ’(’, ’n’, ’o’, ’ ’, ’)’],
[’G’, ’U’, ’G’, ’1’, ’ ’, ’B’, ’A’, ’R’, ’R’, ’ ’, ’ ’, ’1’],
[’G’, ’U’, ’G’, ’2’, ’ ’, ’B’, ’A’, ’R’, ’R’, ’ ’, ’ ’, ’2’], [’G’, ’U’, ’G’, ’3’, ’ ’, ’B’, ’A’, ’R’, ’R’, ’ ’, ’ ’, ’3’],
[’9’, ’9’, ’9’, ’9’, ’9’],
[’E’, ’X’, ’R’, ’S’, ’ ’, ’Q’, ’L’, ’I’, ’M’, ’ ’, ’j’, ’u’, ’m’, ’p’],
[’F’, ’I’, ’M’]]

B.2.3 – DBAR

[[’1’, ’0’, ’L’, ’2’, ’0’, ’Barra1’, ’1’, ’1075’, ’0.’, ’142.5’, ’10.88’, ’-130.’, ’130.4’, ’0.’, ’0.’, ’0.’, ’0.’, ’1’, ’1000’],
[’2’, ’0’, ’L’, ’1’, ’0’, ’Barra2’, ’1’, ’1075’, ’-1.8’, ’90.’, ’-2.59’, ’-101.’, ’101.2’, ’7’, ’0.’, ’0.’, ’0.’, ’2’, ’1000’],
[’3’, ’0’, ’L’, ’1’, ’0’, ’Barra3’, ’1’, ’1075’, ’-1.4’, ’85.’, ’-13.7’, ’-67.4’, ’67.4’, ’9’, ’0.’, ’0.’, ’0.’, ’3’, ’1000’],
[’4’, ’0’, ’L’, ’0’, ’0’, ’Barra4’, ’1’, ’1072’, ’-4.1’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’4’, ’1000’],
[’5’, ’0’, ’L’, ’0’, ’0’, ’Barra5’, ’1’, ’1050’, ’-7.7’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’125.’, ’50.’, ’0.’, ’4’, ’1000’],
[’6’, ’0’, ’L’, ’0’, ’0’, ’Barra6’, ’1’, ’1065’, ’-6.7’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’90.’, ’30.’, ’0.’, ’4’, ’1000’],
[’7’, ’0’, ’L’, ’0’, ’0’, ’Barra7’, ’1’, ’1078’, ’-4.6’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’4’, ’1000’],
[’8’, ’0’, ’L’, ’0’, ’0’, ’Barra8’, ’1’, ’1069’, ’-6.3’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’100.’, ’35.’, ’0.’, ’4’, ’1000’],
[’9’, ’0’, ’L’, ’0’, ’0’, ’Barra9’, ’1’, ’1083’, ’-3.9’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’0.’, ’4’, ’1000’],
[’10’, ’0’, ’L’, ’2’, ’0’, ’Barra10’, ’1’, ’1075’, ’0.’, ’0.000’, ’0.000’, ’-9999’, ’99999’, ’0.’, ’0.’,

’0.’, ’0.’, ’5’, ’1000’]]

B.2.4 – DLIN

[["1","0","0","0","4","1","0","T","0.","5.76","0","1.0","0","0","0","0","250","250","0"],
["10","0","0","0","0","1","1","0","T","0.01","0","0","0","0","0","0","999","999","0"],
["2","0","0","0","7","1","0","T","0.","6.25","0","1.0","0","0","0","0","200","200","0"],
["3","0","0","0","9","1","0","T","0.","5.86","0","1.0","0","0","0","0","300","300","0"],
["4","0","0","0","5","1","0","T","1.","8.5","17.6","0","0","0","0","0","300","300","0"],
["4","0","0","0","6","1","0","T","1.7","9.2","15.8","0","0","0","0","0","200","200","0"],
["6","0","0","0","9","1","0","T","3.9","17.","35.8","0","0","0","0","0","200","200","0"],
["7","0","0","0","5","1","0","T","3.2","16.1","30.6","0","0","0","0","0","300","300","0"],
["7","0","0","0","8","1","0","T",".85","7.2","14.9","0","0","0","0","0","300","300","0"],
["8","0","0","0","9","1","0","T","1.19","10.08","20.9","0","0","0","0","0","300","300","0"]]

B.2.5 – Potências dos equipamentos

CIRCUIT ELEMENT POWER FLOW
(Power Flow into element from indicated Bus)
Power Delivery Elements
Bus Phase kW +j kvar kVA PF
ELEMENT = "Vsource.SOURCE"1 1 -47813.6 +j -3376.8 47932.7 0.9975 1 2 -47813.6 +j -3376.8 47932.7 0.9975 1 3 -47813.6 +j -3376.8 47932.7 0.9975 TERMINAL TOTAL-143440.8 +j -10130.5 143798.1 0.9975 1 0 0.0 +j 0.0 0.0 1.0000 1 0 0.0 +j 0.0 0.0 1.0000 1 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 0.0 +j 0.0 0.0 1.0000
ELEMENT = "Transformer.TR0"1 1 47813.6 +j 3376.8 47932.7 0.9975 1 2 47813.6 +j 3376.8 47932.7 0.9975 1 3 47813.6 +j 3376.8 47932.7 0.9975 1 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL143440.8 +j 10130.5 143798.1 0.9975 4 1 -47713.9 +j -1941.3 47753.4 0.9992 4 2 -47713.9 +j -1941.3 47753.4 0.9992 4 3 -47713.9 +j -1941.3 47753.4 0.9992 4 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL-143141.8 +j -5823.9 143260.2 0.9992
ELEMENT = "Transformer.TR1"2 1 30003.7 +j -863.4 30016.2 -0.9996 2 2 30003.7 +j -863.4 30016.2 -0.9996 2 3 30003.7 +j -863.4 30016.2 -0.9996 2 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 90011.2 +j -2590.3 90048.5 -0.9996 7 1 -29955.2 +j 1621.2 29999.1 -0.9985 7 2 -29955.2 +j 1621.2 29999.1 -0.9985 7 3 -29955.2 +j 1621.2 29999.1 -0.9985 7 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL-89865.7 +j 4863.7 89997.2 -0.9985
ELEMENT = "Transformer.TR2"3 1 28336.8 +j -4567.2 28702.5 -0.9873 3 2 28336.8 +j -4567.2 28702.5 -0.9873 3 3 28336.8 +j -4567.2 28702.5 -0.9873 3 0 0.0 +j 0.0 0.0

1.0000 TERMINAL TOTAL 85010.3 +j -13701.6 86107.4 -0.9873 9 1 -28305.8 +j 5020.8 28747.7 -0.9846 9 2 -28305.8 +j 5020.8 28747.7 -0.9846 9 3 -28305.8 +j 5020.8 28747.7 -0.9846 9 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL-84917.5 +j 15062.5 86243.0 -0.9846

ELEMENT = "Line.L0"4 1 28574.8 +j 12676.7 31260.5 0.9141 4 2 28574.8 +j 12676.7 31260.5 0.9141 4 3 28574.8 +j 12676.7 31260.5 0.9141 TERMINAL TOTAL 85724.5 +j 38030.2 93781.6 0.9141 5 1 -28307.9 +j -10407.4 30160.4 0.9386 5 2 -28307.9 +j -10407.4 30160.4 0.9386 5 3 -28307.9 +j -10407.4 30160.4 0.9386 TERMINAL TOTAL-84923.6 +j -31222.2 90481.1 0.9386

ELEMENT = "Line.L1"4 1 19139.1 +j 1489.8 19197.0 0.9970 4 2 19139.1 +j 1489.8 19197.0 0.9970 4 3 19139.1 +j 1489.8 19197.0 0.9970 TERMINAL TOTAL 57417.3 +j 4469.3 57591.0 0.9970 6 1 -18967.9 +j -563.5 18976.3 0.9996 6 2 -18967.9 +j -563.5 18976.3 0.9996 6 3 -18967.9 +j -563.5 18976.3 0.9996 TERMINAL TOTAL-56903.8 +j -1690.5 56928.9 0.9996

ELEMENT = "Line.L2"6 1 -11034.9 +j 3366.7 11537.0 -0.9565 6 2 -11034.9 +j 3366.7 11537.0 -0.9565 6 3 -11034.9 +j 3366.7 11537.0 -0.9565 TERMINAL TOTAL-33104.6 +j 10100.2 34611.1 -0.9565 9 1 11180.0 +j -2734.1 11509.5 -0.9714 9 2 11180.0 +j -2734.1 11509.5 -0.9714 9 3 11180.0 +j -2734.1 11509.5 -0.9714 TERMINAL TOTAL 33540.1 +j -8202.2 34528.4 -0.9714

ELEMENT = "Line.L3"7 1 13567.4 +j 7289.9 15401.8 0.8809 7 2 13567.4 +j 7289.9 15401.8 0.8809 7 3 13567.4 +j 7289.9 15401.8 0.8809 TERMINAL TOTAL 40702.1 +j 21869.7 46205.5 0.8809 5 1 -13362.9 +j -6260.9 14756.8 0.9055 5 2 -13362.9 +j -6260.9 14756.8 0.9055 5 3 -13362.9 +j -6260.9 14756.8 0.9055 TERMINAL TOTAL-40088.6 +j -18782.6 44270.5 0.9055

ELEMENT = "Line.L4"7 1 16387.9 +j 7976.2 18225.8 0.8992 7 2 16387.9 +j 7976.2 18225.8 0.8992 7 3 16387.9 +j 7976.2 18225.8 0.8992 TERMINAL TOTAL 49163.6 +j 23928.5 54677.5 0.8992 8 1 -16311.8 +j -7331.8 17883.8 0.9121 8 2 -16311.8 +j -7331.8 17883.8 0.9121 8 3 -16311.8 +j -7331.8 17883.8 0.9121 TERMINAL TOTAL-48935.4 +j -21995.3 53651.3 0.9121

ELEMENT = "Line.L5"8 1 -17026.0 +j 3132.1 17311.7 -0.9835 8 2 -17026.0 +j 3132.1 17311.7 -0.9835 8 3 -17026.0 +j 3132.1 17311.7 -0.9835 TERMINAL TOTAL-51078.0 +j 9396.4 51935.1 -0.9835 9 1 17125.8 +j -2286.8 17277.8 -0.9912 9 2 17125.8 +j -2286.8 17277.8 -0.9912 9 3 17125.8 +j -2286.8 17277.8 -0.9912 TERMINAL TOTAL 51377.4 +j -6860.3 51833.4 -0.9912

ELEMENT = "Capacitor.CAP0"4 1 0.0 +j -6442.0 6442.0 1.0000 4 2 0.0 +j -6442.0 6442.0 -0.0000 4 3 0.0 +j -6442.0 6442.0 1.0000 TERMINAL TOTAL 0.0 +j -19326.1 19326.1 -0.0000 4 0 0.0 +j 0.0 0.0 1.0000 4 0 0.0 +j 0.0 0.0 1.0000 4 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 0.0 +j 0.0 0.0 1.0000

ELEMENT = "Capacitor.CAP1"4 1 0.0 +j -5783.2 5783.2 1.0000 4 2 0.0 +j -5783.2 5783.2 1.0000 4 3 0.0 +j -5783.2 5783.2 1.0000 TERMINAL TOTAL 0.0 +j -17349.6 17349.6

1.0000 4 0 0.0 +j 0.0 0.0 1.0000 4 0 0.0 +j 0.0 0.0 1.0000 4 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 0.0 +j 0.0 0.0 1.0000

ELEMENT = "Capacitor.CAP2"6 1 0.0 +j -12804.1 12804.1 1.0000 6 2 -0.0 +j -12804.1 12804.1 0.0000 6 3 -0.0 +j -12804.1 12804.1 0.0000 TERMINAL TOTAL -0.0 +j -38412.4 38412.4 0.0000 6 0 0.0 +j 0.0 0.0 1.0000 6 0 0.0 +j 0.0 0.0 1.0000 6 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 0.0 +j 0.0 0.0 1.0000

ELEMENT = "Capacitor.CAP3"7 1 0.0 +j -11357.2 11357.2 1.0000 7 2 0.0 +j -11357.2 11357.2 1.0000 7 3 0.0 +j -11357.2 11357.2 1.0000 TERMINAL TOTAL 0.0 +j -34071.5 34071.5 1.0000 7 0 0.0 +j 0.0 0.0 1.0000 7 0 0.0 +j 0.0 0.0 1.0000 7 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 0.0 +j 0.0 0.0 1.0000

ELEMENT = "Capacitor.CAP4"7 1 -0.0 +j -5530.1 5530.1 0.0000 7 2 0.0 +j -5530.1 5530.1 1.0000 7 3 0.0 +j -5530.1 5530.1 1.0000 TERMINAL TOTAL -0.0 +j -16590.4 16590.4 0.0000 7 0 0.0 +j 0.0 0.0 1.0000 7 0 0.0 +j 0.0 0.0 1.0000 7 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 0.0 +j 0.0 0.0 1.0000

ELEMENT = "Capacitor.CAP5"8 1 0.0 +j -7468.6 7468.6 -0.0000 8 2 -0.0 +j -7468.6 7468.6 0.0000 8 3 -0.0 +j -7468.6 7468.6 0.0000 TERMINAL TOTAL -0.0 +j -22405.7 22405.7 0.0000 8 0 0.0 +j 0.0 0.0 1.0000 8 0 0.0 +j 0.0 0.0 1.0000 8 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 0.0 +j 0.0 0.0 1.0000

= =

Power Conversion Elements

Bus Phase kW +j kvar kVA PF

ELEMENT = "Load.C4"5 1 41668.3 +j 16668.3 44878.5 0.9285 5 2 41668.3 +j 16668.3 44878.5 0.9285 5 3 41668.3 +j 16668.3 44878.5 0.9285 5 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 125005.0 +j 50004.8 134635.6 0.9285

ELEMENT = "Load.C5"6 1 30001.2 +j 10000.9 31624.2 0.9487 6 2 30001.2 +j 10000.9 31624.2 0.9487 6 3 30001.2 +j 10000.9 31624.2 0.9487 6 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 90003.6 +j 30002.6 94872.6 0.9487

ELEMENT = "Load.C7"8 1 33335.4 +j 11667.8 35318.3 0.9439 8 2 33335.4 +j 11667.8 35318.3 0.9439 8 3 33335.4 +j 11667.8 35318.3 0.9439 8 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL 100006.1 +j 35003.3 105954.9 0.9439

ELEMENT = "Generator.G1"2 1 -30001.8 +j 863.5 30014.2 -0.9996 2 2 -30001.8 +j 863.5 30014.2 -0.9996 2 3 -30001.8 +j 863.5 30014.2 -0.9996 2 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL -90005.3 +j 2590.6 90042.6 -0.9996

ELEMENT = "Generator.G2"3 1 -28335.0 +j 4567.1 28700.7 -0.9873 3 2 -28335.0 +j 4567.1 28700.7 -0.9873 3 3 -28335.0 +j 4567.1 28700.7 -0.9873 3 0 0.0 +j 0.0 0.0 1.0000 TERMINAL TOTAL -85004.9 +j 13701.3 86102.0 -0.9873

Total Circuit Losses = 3428.5 +j 26982.1

B.2.6 – Tensões de fase e pu dos equipamentos

NODE-GROUND VOLTAGES BY CIRCUIT ELEMENT

Power Delivery Elements

Bus (node ref) Phase Magnitude, kV (pu) Angle

ELEMENT = "Vsource.SOURCE"1 (1) 1 0.60727 (1.052) /_ -3.5 1 (2) 2 0.60727 (1.052) /_ -123.5 1 (3) 3 0.60727 (1.052) /_ 116.5 ———— 1 (0) 0 0 (0) /_ 0.0 1 (0) 0 0 (0) /_ 0.0 1 (0) 0 0 (0) /_ 0.0

ELEMENT = "Transformer.TR0"1 (1) 1 0.60727 (1.052) /_ -3.5 1 (2) 2 0.60727 (1.052) /_ -123.5 1 (3) 3 0.60727 (1.052) /_ 116.5 1 (0) 0 0 (0) /_ 0.0 ———— 4 (4) 1 0.605 (1.048) /_ -5.2 4 (5) 2 0.605 (1.048) /_ -125.2 4 (6) 3 0.605 (1.048) /_ 114.8 4 (0) 0 0 (0) /_ 0.0

ELEMENT = "Transformer.TR1"2 (7) 1 0.60957 (1.056) /_ -4.3 2 (8) 2 0.60957 (1.056) /_ -124.3 2 (9) 3 0.60957 (1.056) /_ 115.7 2 (0) 0 0 (0) /_ 0.0 ———— 7 (10) 1 0.60922 (1.055) /_ -5.8 7 (11) 2 0.60922 (1.055) /_ -125.8 7 (12) 3 0.60922 (1.055) /_ 114.2 7 (0) 0 0 (0) /_ 0.0

ELEMENT = "Transformer.TR2"3 (13) 1 0.59568 (1.032) /_ -3.8 3 (14) 2 0.59568 (1.032) /_ -123.8 3 (15) 3 0.59568 (1.032) /_ 116.2 3 (0) 0 0 (0) /_ 0.0 ———— 9 (16) 1 0.59662 (1.033) /_ -4.7 9 (17) 2 0.59662 (1.033) /_ -124.7 9 (18) 3 0.59662 (1.033) /_ 115.3 9 (0) 0 0 (0) /_ 0.0

ELEMENT = "Line.L0"4 (4) 1 0.605 (1.048) /_ -5.2 4 (5) 2 0.605 (1.048) /_ -125.2 4 (6) 3 0.605 (1.048) /_ 114.8 ———— 5 (19) 1 0.58371 (1.011) /_ -8.9 5 (20) 2 0.58371 (1.011) /_ -128.9 5 (21) 3 0.58371 (1.011) /_ 111.1

ELEMENT = "Line.L1"4 (4) 1 0.605 (1.048) /_ -5.2 4 (5) 2 0.605 (1.048) /_ -125.2 4 (6) 3 0.605 (1.048) /_ 114.8 ———— 6 (22) 1 0.59804 (1.036) /_ -7.9 6 (23) 2 0.59804 (1.036) /_ -127.9 6 (24) 3 0.59804 (1.036) /_ 112.1

ELEMENT = "Line.L2"6 (22) 1 0.59804 (1.036) /_ -7.9 6 (23) 2 0.59804 (1.036) /_ -127.9 6 (24) 3 0.59804 (1.036) /_ 112.1 ———— 9 (16) 1 0.59662 (1.033) /_ -4.7 9 (17) 2 0.59662 (1.033) /_ -124.7 9 (18) 3 0.59662 (1.033) /_ 115.3

ELEMENT = "Line.L3"7 (10) 1 0.60922 (1.055) /_ -5.8 7 (11) 2 0.60922 (1.055) /_ -125.8 7 (12) 3 0.60922 (1.055) /_ 114.2 ———— 5 (19) 1 0.58371 (1.011) /_ -8.9 5 (20) 2 0.58371 (1.011) /_ -128.9 5 (21) 3 0.58371 (1.011) /_ 111.1

ELEMENT = "Line.L4"7 (10) 1 0.60922 (1.055) /_ -5.8 7 (11) 2 0.60922 (1.055) /_ -125.8 7 (12) 3 0.60922 (1.055) /_ 114.2 ———— 8 (25) 1 0.59779 (1.035) /_ -7.5 8 (26) 2 0.59779 (1.035) /_ -127.5 8 (27) 3 0.59779 (1.035) /_ 112.5

ELEMENT = "Line.L5"8 (25) 1 0.59779 (1.035) /_ -7.5 8 (26) 2 0.59779 (1.035) /_ -127.5 8 (27) 3 0.59779 (1.035) /_ 112.5 ———— 9 (16) 1 0.59662 (1.033) /_ -4.7 9 (17) 2 0.59662 (1.033) /_ -124.7 9 (18) 3 0.59662 (1.033) /_ 115.3

ELEMENT = "Capacitor.CAP0"4 (4) 1 0.605 (1.048) /_ -5.2 4 (5) 2 0.605 (1.048) /_ -125.2 4 (6) 3 0.605 (1.048) /_ 114.8 ———— 4 (0) 0 0 (0) /_ 0.0 4 (0) 0 0 (0) /_ 0.0

4 (0) 0 0 (0) /_ 0.0

ELEMENT = "Capacitor.CAP1"4 (4) 1 0.605 (1.048) /_ -5.2 4 (5) 2 0.605 (1.048) /_ -125.2 4 (6) 3 0.605 (1.048) /_ 114.8 ———— 4 (0) 0 0 (0) /_ 0.0 4 (0) 0 0 (0) /_ 0.0 4 (0) 0 0 (0) /_ 0.0

ELEMENT = "Capacitor.CAP2"6 (22) 1 0.59804 (1.036) /_ -7.9 6 (23) 2 0.59804 (1.036) /_ -127.9 6 (24) 3 0.59804 (1.036) /_ 112.1 ———— 6 (0) 0 0 (0) /_ 0.0 6 (0) 0 0 (0) /_ 0.0 6 (0) 0 0 (0) /_ 0.0

ELEMENT = "Capacitor.CAP3"7 (10) 1 0.60922 (1.055) /_ -5.8 7 (11) 2 0.60922 (1.055) /_ -125.8 7 (12) 3 0.60922 (1.055) /_ 114.2 ———— 7 (0) 0 0 (0) /_ 0.0 7 (0) 0 0 (0) /_ 0.0 7 (0) 0 0 (0) /_ 0.0

ELEMENT = "Capacitor.CAP4"7 (10) 1 0.60922 (1.055) /_ -5.8 7 (11) 2 0.60922 (1.055) /_ -125.8 7 (12) 3 0.60922 (1.055) /_ 114.2 ———— 7 (0) 0 0 (0) /_ 0.0 7 (0) 0 0 (0) /_ 0.0 7 (0) 0 0 (0) /_ 0.0

ELEMENT = "Capacitor.CAP5"8 (25) 1 0.59779 (1.035) /_ -7.5 8 (26) 2 0.59779 (1.035) /_ -127.5 8 (27) 3 0.59779 (1.035) /_ 112.5 ———— 8 (0) 0 0 (0) /_ 0.0 8 (0) 0 0 (0) /_ 0.0 8 (0) 0 0 (0) /_ 0.0

= =

Power Conversion Elements

Bus (node ref) Phase Magnitude, kV (pu) Angle

ELEMENT = "Load.C4"5 (19) 1 0.58371 (1.011) /_ -8.9 5 (20) 2 0.58371 (1.011) /_ -128.9 5 (21) 3 0.58371 (1.011) /_ 111.1 5 (0) 0 0 (0) /_ 0.0

ELEMENT = "Load.C5"6 (22) 1 0.59804 (1.036) /_ -7.9 6 (23) 2 0.59804 (1.036) /_ -127.9 6 (24) 3 0.59804 (1.036) /_ 112.1 6 (0) 0 0 (0) /_ 0.0

ELEMENT = "Load.C7"8 (25) 1 0.59779 (1.035) /_ -7.5 8 (26) 2 0.59779 (1.035) /_ -127.5 8 (27) 3 0.59779 (1.035) /_ 112.5 8 (0) 0 0 (0) /_ 0.0

ELEMENT = "Generator.G1"2 (7) 1 0.60957 (1.056) /_ -4.3 2 (8) 2 0.60957 (1.056) /_ -124.3 2 (9) 3 0.60957 (1.056) /_ 115.7 2 (0) 0 0 (0) /_ 0.0

ELEMENT = "Generator.G2"3 (13) 1 0.59568 (1.032) /_ -3.8 3 (14) 2 0.59568 (1.032) /_ -123.8 3 (15) 3 0.59568 (1.032) /_ 116.2 3 (0) 0 0 (0) /_ 0.0

NUP: 23081.063610/2022-81 **Prioridade:** Normal

Homologação de ata de defesa de TCC e estágio de graduação

125.322 - Bancas examinadoras de TCC: indicação e atuação

COMPONENTE

Ordem	Descrição	Nome do arquivo
11	Trabalho de conclusão de curso (TCC) (125.32)	TCC.pdf

Assinaturas

20/07/2022 08:32:00

DANIEL PINHEIRO BERNARDON (PROFESSOR DO MAGISTÉRIO SUPERIOR)
07.37.00.00.0.0 - DEPARTAMENTO DE ELETROMECÂNICA E SISTEMAS DE POTÊNCIA - DESP