



THE OPEN DISTRIBUTION SYSTEM SIMULATOR
OPENDSS

Introdução ao OpenDSS

Autor:

Jason Sexauer, OpenDSS User

Tradução e Edição: Paulo Radatz e Celso Rocha

26 de julho de 2016

1 License

Copyright (c) 2008-2012, Electric Power Research Institute, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the Electric Power Research Institute, Inc., nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY Electric Power Research Institute, Inc., "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL Electric Power Research Institute, Inc., BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Sumário

1	License	1
2	O que é o <i>software</i> OpenDSS?	5
2.1	História do <i>Software</i> OpenDSS	6
2.2	O Propósito desse Documento	7
3	Modelos Básicos do OpenDSS	7
3.1	Barra	8
3.2	Terminal	8
3.3	Elementos de Transporte de Energia (PD)	9
3.4	Elementos de Conversão de Energia (PC)	9
3.5	Elementos de Suporte	10
4	O OpenDSS <i>Standalone</i> e sua Interface	11
4.1	Utilizando a GUI do OpenDSS	12
4.2	Passos Comuns	13
5	A Linguagem de Programação do OpenDSS	14
5.1	Sintaxe da Linguagem de Programação do OpenDSS	14
5.2	Verbo	15
5.3	Parâmetros	15
5.4	Comentários	17
5.5	Multi Linhas	17
5.6	Incluindo Arquivos Externos	18
5.7	Exemplo	18
6	Introdução à COM Interface	21
6.1	Começando com a interface COM	22
6.2	Interface <i>Text</i>	23
6.3	Interface <i>Circuit</i>	23
6.4	Interface <i>Solution</i>	25
7	Controlando o OpenDSS via COM	26
7.1	Exemplo de Controle Utilizando o VBA Excel	26
7.2	Exemplo de Controle Utilizando o MATLAB	28
7.3	Exemplo de Controle Utilizando o Python	29
8	Recursos Adicionais	30
8.1	Onde ir a partir daqui?	31
8.2	Recursos de Referência	32

8.3	Ajuda Adicional	32
8.4	Contato	33

2 O que é o *software* OpenDSS?

O *Open Distribution System Simulator* (OpenDSS) é um *software* de simulação para sistemas elétricos de distribuição de energia elétrica. O OpenDSS é implementado em duas versões. A primeira corresponde a um programa executável autônomo e a segunda a um servidor COM (*Component Object Model*) implementado a partir de uma DLL. Na versão autônoma há uma interface de usuário que permite a descrição de circuitos, a solução e a visualização de resultados. No segundo caso, o OpenDSS pode ser controlado por uma variedade de plataformas de softwares existentes.

O programa suporta a maioria das análises em regime permanente senoidal (RMS) comumente realizadas para análise e planejamento de sistemas de distribuição de energia elétrica. Além disso, o OpenDSS é capaz de realizar novos tipos de análises que são necessárias para atender as necessidades futuras relacionadas as Redes Elétricas Inteligentes (*Smart Grids*). Muitos dos recursos encontrados no programa foram originalmente destinados a dar suporte às necessidades das análises em que há geração distribuída (GD). Outros recursos dão suporte à análise de eficiência energética do transporte de energia e análise harmônica. Por fim, o OpenDSS foi projetado para ser indefinidamente expansível de tal modo que possa ser facilmente modificado para atender as necessidades futuras ou para atender as necessidades específicas dos usuários.

Outra característica principal do OpenDSS está nos seus modos de solução *quasi-static*, que permitem a realização de simulações sequenciais no tempo. Assim, é possível analisar como os sistemas se comportam durante um intervalo de tempo definido.

A seguir são apresentados os principais modos de solução presentes no OpenDSS:

- Fluxo de potência instantâneo (*SnapShot*)
- Fluxo de potência diário
- Fluxo de potência anual
- Análise harmônica
- Análise dinâmica
- Estudo de curto circuitos

Esses modos de solução foram adicionados ao longo do tempo para atender às necessidades de análise de projetos específicos nos quais os autores estavam envolvidos. No entanto, o OpenDSS foi concebido sabendo que os desenvolvedores nunca seriam capazes de prever todas as necessidades dos usuários. Portanto, uma interface COM

foi implementada através da versão DLL do programa para permitir que usuários mais experientes possam utilizar os recursos do *software* para realizar novos tipos de estudos. Através da interface COM, o usuário é capaz de projetar e executar modos de solução próprios através de um programa externo e realizar as funções do simulador, incluindo a definição dos dados dos modelos. Em outras palavras, o OpenDSS pode ser implementado inteiramente de modo independente de qualquer banco de dados fixo ou de um circuito definido em um arquivo de texto fixo.

Por exemplo, o OpenDSS pode ser controlado por programas como Python, *MATrix LABoratory* (MATLAB), por ferramentas da MS *Office*, com destaque para o *Visual Basic for Applications* (VBA), dentre outros programas.

2.1 História do *Software* OpenDSS

O desenvolvimento do DSS iniciou-se em 1997 na *Electrotek Concepts, Inc.* por Roger Dugan e Thomas McDemontt.

As principais finalidades do *software* naquele momento consistiam em dar apoio às análises de redes com a presença de GD, ser uma ferramenta flexível capaz de realizar análises que outras ferramentas não realizavam e ter um desenvolvimento constante com o objetivo de criar soluções para novos desafios relativos aos sistemas de distribuição.

Em 2004, o DSS foi comprado pelo EPRI e, em 2008, o EPRI apresentou sob licença de código aberto o chamado *software* OpenDSS, a fim de contribuir para o desenvolvimento da área de *Smart Grids*.

Como consequência do seu desenvolvimento inicial, o OpenDSS está mais perto de ser um programa de análise de fluxo harmônico do que um programa fluxo de potência típico. Outra característica importante do OpenDSS é a forma que o *software* trata as barras. Enquanto na maioria dos programas de fluxo de potência as barras são os elementos centrais na qual todo o sistema é construído, no OpenDSS as barras são criados dinamicamente conforme necessário. Essas duas características podem parecer diferentes em uma ferramenta que será usada principalmente para estudos de fluxo de potência, entretanto ambas dão a flexibilidade necessária para se modelar todos os tipos de carga e definir configurações incomuns de elementos como os transformadores. Resumindo, é mais fácil fazer um programa de análise de fluxo harmônico resolver o problema de fluxo de potência do que o contrário.

2.2 O Propósito desse Documento

O objetivo deste documento é servir de auxílio para usuários iniciantes. Embora inclua grande parte das informações do Manual do OpenDSS, esse documento não se destina ao entendimento profundo do *software*. Conforme necessário, durante esse documento serão fornecidas referências de onde o leitor interessado pode encontrar mais informações. Além disso, há dois tutoriais em português disponíveis no YouTube. O primeiro trata dos conceitos básicos do *software*, incluindo a definição dos principais elementos, a solução no modo *snapshot* e como visualizar resultados na versão autônoma do *software*, enquanto que o segundo foca essencialmente na utilização do *software* através da interface COM. O primeiro tutorial pode ser acessado clicando-se [aqui](#) e o segundo tutorial pode ser acessado clicando-se [aqui](#).

3 Modelos Básicos do OpenDSS

O OpenDSS modela os seus sistemas elétricos para simulações na condição de regime permanente senoidal (RPS), ou seja, consiste em uma ferramenta de cálculo algébrico, no domínio fasorial. Sobreposto ao sistema elétrico, encontra-se uma rede de comunicação que interconecta controles e medições sobre os elementos elétricos.

Os blocos básicos para a construção de um circuito são os elementos de transporte de energia (Power Delivery - PD) e os elementos de conversão de energia (Power Conversion - PC). Os principais PDs são as linhas e os transformadores, e os principais PCs são as cargas e os geradores.

Os elementos de controle, de medição e elementos gerais, como curvas de carga, são chamados de elementos de suporte e podem ser criados para refinar os modelos dos PDs e dos PCs. Mais informações sobre todos esses modelos estão disponíveis no Manual do OpenDSS.

Para conectar os elementos elétricos, PDs e PCS, são utilizados barras que contém nós, entretanto essas barras são criadas como uma consequência da criação de um elemento. Isto é uma mudança de paradigma significativa sobre os programas de fluxo de potência tradicionais, nos quais as barras são elementos centrais. Clique [aqui](#) para assistir uma breve explicação dessa forma de construção dos circuitos utilizada pelo OpenDSS.

3.1 Barra

A barra no OpenDSS é um elemento de circuito que contém nós, como apresentado na Figura 1. A principal função desse elemento é conectar em seus nós os terminais dos elementos elétricos, que podem ser os PDs ou os PCs. Toda barra apresenta um nó de referência, nó 0, que apresenta tensão nula.

Por fim, uma característica muito importante do OpenDSS é que as barras são apenas criadas a partir da inclusão de elementos elétricos no sistema. Por exemplo, duas barras podem ser criadas quando se declara uma linha e uma barra pode ser criada quando se declara uma carga. O verbo “poder” é utilizado, pois há a possibilidade de se criar uma linha entre duas barras que já foram definidas devido a criação de outros elementos.



Figura 1: Barra com N nós

3.2 Terminal

Os elementos elétricos possuem um ou mais terminais que podem conter diversos conectores. Cada terminal deve ser conectado em apenas uma barra. A Figura 2 apresenta um elemento com um terminal que contém N conectores.

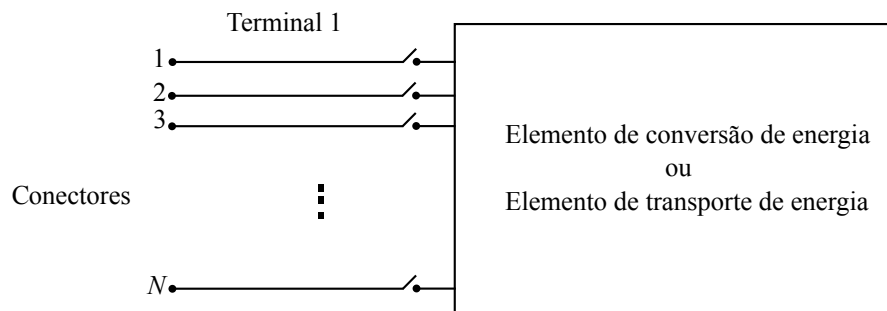


Figura 2: Terminal com N conectores

3.3 Elementos de Transporte de Energia (PD)

Os elementos de transporte de energia, Figura 3, possuem um ou mais terminais como, por exemplo:

- Uma linha à 4 fios apresenta dois terminais com 4 conectores cada;
- Um transformador monofásico de três enrolamentos apresenta 3 terminais com 2 conectores cada;
- Um banco de capacitores trifásico apresenta apenas um terminal com 3 conectores.

A função básica dos elementos de transporte de energia é levar energia de um ponto para o outro. Esses elementos são caracterizados pela sua matriz de admitância nodal.

Vale ressaltar que para bancos de capacitores e reatores conectados em *shunt* a ideia de transportar a energia de um ponto para o outro não é válida, entretanto, esse elementos também são definidos no OpenDSS como elementos de transporte de energia.

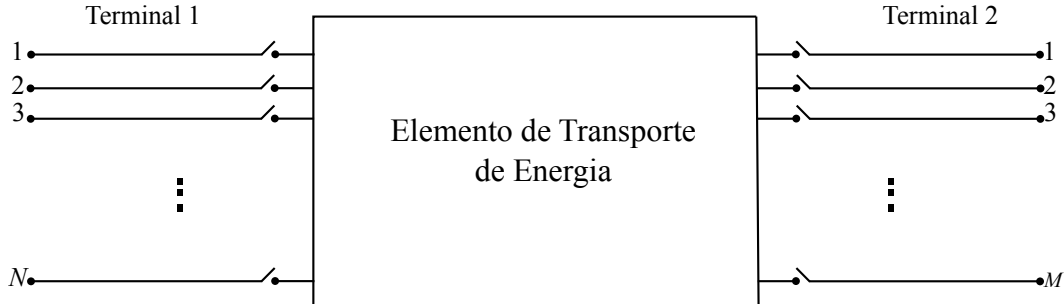


Figura 3: Elemento de transporte de energia com dois terminais, o primeiro com N conectores e o segundo com M conectores

3.4 Elementos de Conversão de Energia (PC)

Os elementos de conversão de energia convertem energia elétrica para outra forma de energia, ou vice-versa. Eles são basicamente os geradores, cargas, acumuladores de energia, dentre outros. Em geral, apresentam apenas um terminal com N conectores, conforme apresentado na Figura 4.

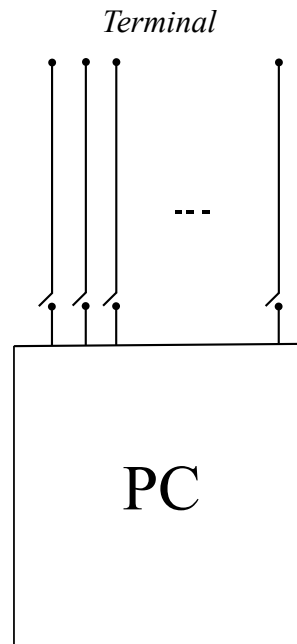


Figura 4: Elemento de conversão de energia

3.5 Elementos de Suporte

O OpenDSS disponibiliza uma variedade de elementos de suporte, que podem ser associados aos PDs e aos PCs. Os elementos de suporte fornecem a possibilidade de: facilitar a definição de parâmetros para os PDs e os PCs; associar funcionalidades de controle à alguns elementos; monitorar ou fornecer dados e resultados para as análises temporais.

A seguir são apresentados os principais elementos de suporte utilizados:

- LineCode: Armazena as informações elétricas de uma linha específica, ou seja, o arranjo. Uma breve explicação sobre esse elemento pode ser encontrada clicando-se [aqui](#).
- LineGeometry e WireData: Armazenam as informações geométricas dos postes e das linhas. O OpenDSS utiliza essas informações para construir a matriz de admitância nodal da linha em $S/Unidade de Comprimento$.
- LoadShape: Corresponde à curva de carga associada a um consumidor ou à curva de geração associada a um gerador. Esse elemento de suporte é utilizado quando se deseja realizar simulações sequenciais no tempo.
- Spectrum: A inclusão de um espectro harmônico a um elemento de Conversão

de Energia.

- **EnergyMeter:** Esse elemento de suporte corresponde a um medidor que fornece diversos resultados sobre a sua área de medição.
- **Monitor:** Os monitores são instalados em pontos específicos para fornecerem resultados elétricos escolhidos.
- **CapControl:** Esse elemento de suporte corresponde a um elemento de controle que tem como objetivo controlar um capacitor.
- **RegControl:** Esse elemento de suporte corresponde a um elemento de controle que tem como objetivo controlar o tap ou taps de um transformador.

4 O OpenDSS *Standalone* e sua Interface

O OpenDSS *Standalone* é apresentado com uma *Graphical User Interface* (GUI) que proporciona um ambiente estruturado para a criação e análise de sistemas elétricos.

A criação de sistemas elétricos nessa GUI deve ser realizada utilizando linhas de comandos seguindo a linguagem de programação do OpenDSS. Esses comandos são documentados em detalhes no Manual do OpenDSS. **Todos os comandos executados pela GUI possuem um comando correspondente na linguagem de programação do OpenDSS.** Esses comandos podem ser verificados através da **Ajuda** do OpenDSS (Help → OpenDSS Help) ou através da ferramenta de gravação de comandos (Edit → Record Script). Quando ativada, qualquer funcionalidade executada pelo usuário através da GUI é adicionada ao final do código da janela selecionada e no correspondente formato na linguagem de programação do OpenDSS.

Para executar linhas ou fragmentos de linhas de comandos são necessários seguir os seguintes passos:

1. Selecione a linha ou as linhas que se deseja rodar.
2. Clique com o botão direito do mouse e selecione *Do Selected* ou a tecla de atalho Ctrl+D. (A seleção também pode ser executada a partir do menu "Do" ou diretamente através do botão abaixo do menu "Edit")

Os resultados das simulações são fornecidos por meio de arquivos *.csv ou *.txt. O OpenDSS gera alguns relatórios de resultados padrões (ver o menu *Show* e o menu *Export*). Caso o usuário deseje relatórios mais sofisticados, é imprescindível a utilização da interface COM.

4.1 Utilizando a GUI do OpenDSS

Ao abrir o OpenDSS, o usuário encontrará a janela apresentada na Figura 5.

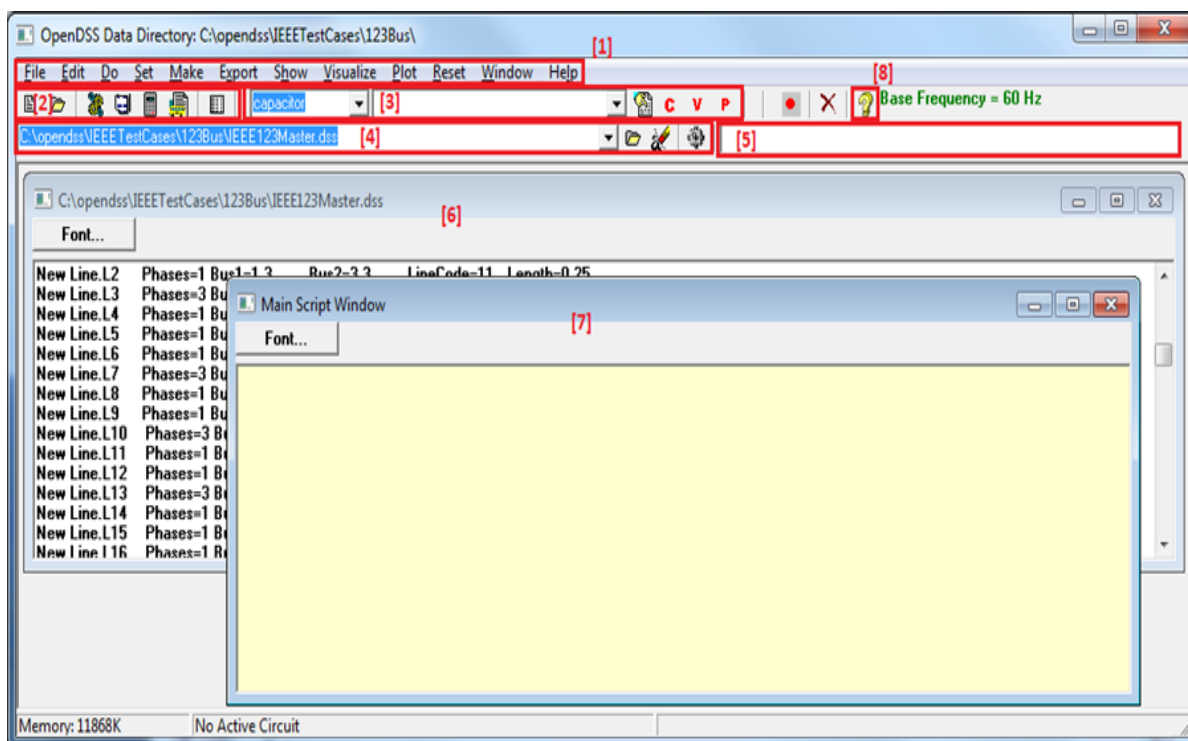


Figura 5: Interface do OpenDSS *Standalone*

Os principais componentes da interface são:

1. Estrutura do Menu:

- Menu *Set*: Permite que se defina qualquer parâmetro de solução.
- Menu *Export*: Permite salvar diversos relatórios de resultados no formato *.csv. Para assistir a um vídeo com uma breve explicação, clique [aqui](#).
- Menu *Show*: Abre o relatório escolhido no editor de texto selecionado. Para assistir ao vídeo com uma breve explicação, clique [aqui](#).
- Menu *Visualize*: Fornece o diagrama do elemento selecionado com a grandeza elétrica escolhida. Para assistir ao vídeo com uma breve explicação, clique [aqui](#).
- Menu *Plot*: Fornece diversos gráficos do sistema elétrico em análise. Para assistir ao vídeo com uma breve explicação, clique [aqui](#).

2. *Barra de Ferramentas*: Fornece acesso direto a alguns comandos importantes do OpenDSS como, por exemplo, *Solve*, *Summary* e *Do Command*.
3. *Ferramenta de Elementos*: Permite que o usuário selecione um tipo de elemento com objetivo de editá-lo ou exibi-lo.
4. *Ferramenta de Script*: Permite que o usuário escolha qual das janelas abertas deverá ser simulada.
5. *Barra de Resultados*: Proporciona uma versão condensada da janela de resultados, que pode ser acessada através de *Show → Results Form*.
6. *Janela de Script Principal*: É uma espécie de bloco de anotações ou janela interativa do OpenDSS *Standalone* que possibilita ao usuário poder digitar linhas de comandos e executá-las através do atalho Ctrl+D. Entretanto, essa janela não pode ser fechada. O seu conteúdo é mantido mesmo após o fechamento do *software*.
7. *Janelas de Scripts*: Tem as mesmas funcionalidades da *Janela de Script Principal*, porém pode ser fechada. Portanto, a inclusão de novas janelas possibilita a edição de arquivos DSS simultaneamente e a sua execução individual através do atalho Ctrl+D.
8. *Botão de Ajuda*: Ajuda o usuário a encontrar os comandos e as suas correspondentes descrições.

4.2 Passos Comuns

A seguir são apresentados os passos comuns realizados em simulações utilizando a GUI do OpenDSS:

1. Definição do circuito: deve-se inserir linhas de comandos para definir linhas, transformadores, cargas, geradores, etc...
2. Configuração das opções de simulação: selecionar o modo de solução (instantâneo, diário, análise harmônica, etc...)
3. Resolver o Fluxo de Potência: selecionar as linhas de comandos e então usar o atalho Ctrl+D.
4. Análise dos resultados:
 - Observando um Elemento em Particular: selecione primeiramente o elemento na *Ferramenta de Elementos* e, por fim, use o menu *Visualize*.

- Observando o Perfil de Tensão: após o fluxo de potência ter sido realizado, digite a linha de comando *plot Profile* e execute esse comando. Vale salientar que só é possível plotar o perfil de tensão do alimentador se um *EnergyMeter* for inserido no seu início.
- Observando Informações das Linhas no Plot do Circuito: para plotar o circuito é necessário utilizar o menu *Plot* juntamente com a definição das coordenadas das barras através do comando *Buscoords* (para mais informações, consulte o Manual do OpenDSS).
- Observando os Relatórios de Resultados: pode-se utilizar os menus *Show* e *Export*.

5 A Linguagem de Programação do OpenDSS

A descrição dos circuitos e a sua resolução são realizados através de linhas de comandos respeitando a linguagem de programação do OpenDSS. Esse conjunto de linhas de comandos podem ser processados pelo OpenDSS de 3 formas ou a combinação dessas:

1. Circuito descrito na *Janela de Script* da GUI do OpenDSS *Standalone*.
2. Circuito descrito através da interface COM.
3. Circuito descrito em um arquivo DSS externo (mais informações, consulte os comandos *Compile* e *Redirect* no Manual do OpenDSS).

Essa característica faz do OpenDSS uma ferramenta facilmente acessível para usuários que querem realizar um rápido estudo em um circuito bem como para aqueles que querem realizar estudos mais complexos. Além disso, o OpenDSS facilita a utilização de diferentes bancos de dados, ou seja, não há a necessidade de se conhecer um banco de dados específico.

É sempre interessante utilizar a **Ajuda** do *software* para se conhecer ou recordar os comandos da linguagem de programação do OpenDSS. Vale salientar que essa linguagem não faz distinção entre letras maiúsculas ou minúsculas.

5.1 Sintaxe da Linguagem de Programação do OpenDSS

Cada comando corresponde a uma linha de texto. Via de regra, cada comando apresenta um **Verbo** e alguns **Parâmetros**.

Existem dois tipos de linhas de comando. A primeira corresponde àquela em que o verbo interage diretamente com um elemento de circuito como, por exemplo:

```
VerboComando ClasseElemento.NomeElemento Param1=Valor1 Param2=Valor2
New Line.Linha Phases=3 Bus1=A Bus2=B length=1 units=km linecode=
ArranjoTeste
```

O segundo tipo de linha de comando corresponde àquela em que o verbo não interage com um elemento, por exemplo:

```
VerboComando Param1=Valor1 Param2=Valor2
Set mode=daily stepsize=1h number=24
```

5.2 Verbo

A Tabela 1 apresenta os verbos mais comuns. Para encontrar todos os verbos disponíveis, clique no botão de **Ajuda** e selecione a opção *Executive*.

Tabela 1: Principais Verbos

Verbo	Descrição
New	Cria um novo elemento de circuito
Edit	Edita um elemento de circuito escolhido
Set	Define as opções de solução como, por exemplo, <i>Mode</i>
Solve	Realiza a solução do circuito definido
Show	Apresenta relatórios de resultados em arquivos *.txt
Export	Salva relatórios de resultados em arquivo *.csv
Plot	Plota os resultados do fluxo de potência no circuito

Caso nenhum verbo seja especificado, o verbo *Edit* é utilizado.

5.3 Parâmetros

Os pares parâmetro/valor podem ser separados através de vírgula ou de espaço em branco. Para valores que requerem delimitadores, podem ser utilizados os apresentados na Tabela 2. Embora esses delimitadores sejam intercambiáveis, é sugerido seguir um padrão para uma melhor estética.

Para o acesso a um membro de uma classe ou um parâmetro (propriedade) de um elemento é utilizado o símbolo de ponto final (.). Todos os elementos devem ser

Tabela 2: Delimitadores da Linguagem de Programação do OpenDSS

Delimitador	Sugestão de Uso
Aspas simples	<i>String</i>
Aspas duplas	<i>String</i>
Parênteses	Lista de valores
Colchetes	Lista de valores
Chaves	Expressão Matemática

especificados através de seu nome completo, a menos que o contexto deixe claro qual é o elemento. Por exemplo, quando especificamos um elemento CapControl, o elemento capacitor já é inferido. Por esse motivo, o nome da classe do elemento não precisa ser incluído, ou seja, se o capacitor *Capacitor.Cap1* deve ser controlado por um elemento *CapControl*, na descrição do elemento *CapControl* deve-se especificar apenas *Cap1*, isto é, *capacitor = Cap1*.

O nome completo de um elemento possui o seguinte formato:

```
ClasseElemento.NomeElemento
Line.Linha
```

Para acessar uma propriedade de um elemento, o seguinte formato deve ser utilizado:

```
ClasseElemento.NomeElemento.PropriedadeNome
Line.Linha.Phases
```

Ao acessar múltiplos parâmetros do mesmo elemento dentro do mesmo comando, os nomes de propriedades adicionais não precisam ser descritos por completo. Por exemplo:

```
Line.L1.Bus1=1, Bus2=5
```

Além disso, os parâmetros também podem ser especificados na sua ordem padrão. Isso é comumente feito quando se especifica a propriedade *elemento* ou *objeto*, pois geralmente essas propriedades são as primeiras a serem descritas. Por fim, a especificação pela ordem e pelo par parâmetro/valor podem ser misturadas, como mostrado abaixo:

```
New EnergyMeter.Feeder Line.L115, terminal=1, enabled=false
```

Ao especificar um novo dispositivo, à muitos dos seus parâmetros são dados valores razoáveis. A documentação completa sobre quais parâmetros devem ser especifica-

dos, quais parâmetros recebem possuem/valores padrões (*default*) e quais são esses valores é fornecida na documentação dos comandos.

Em geral, classes de elementos que possuem as propriedades *element*, *object*, *terminal*, *bus1* ou *bus2* não possuem valores padrão e deve ser especificados no instante da criação do dispositivo.

5.4 Comentários

Comentários podem ser adicionados às linhas de comando utilizando os símbolos (//) ou (!), conforme exemplo a seguir:

```
// Editar o Controle do Regulador de tensao chamado ControleRegulador
RegControl.ControleRegulador.Maxtapchange=1 ! Limite para mudanca de
um tap
```

Para inserir comentários em múltiplas linhas é necessário utilizar (/* ... */), conforme exemplo a seguir:

```
/* Comentando um trecho de codigo
New Monitor.FontePQ Vsource.Fonte 1 mode=1 ppolar=no
New Monitor.FonteVI Vsource.Fonte 1 mode=0 VIpolar=Yes
*/
```

5.5 Multi Linhas

Se um comando precisar de mais de uma linha, o símbolo (~) deve ser utilizado no começo das linhas adicionadas, conforme apresentado a seguir:

```
New Line.L1 Bus1=1, Bus2=2, Length=1
~ units=mi, geometry=3PH.3/0_Horiz
```

Observe que, na realidade, o código acima consiste em dois comandos separados. Como tal, todos os parâmetros não-padrão devem ser definidos na primeira linha de comando. Caso contrário o OpenDSS acusará um erro de sintaxe, como no caso a seguir:

```
// O OpenDSS acusara um erro pois as barras Bus1 e Bus2 nao sao
definidas na primeira linha
New Line.L1 Length=1, units=mi
~ Bus1=1, Bus2=2, geometry=3PH.3/0_Horiz
```

5.6 Incluindo Arquivos Externos

Há duas maneiras de incluir arquivos DSS externos dentro do código analisado pelo OpenDSS:

1. Comando *Compile*: Esse comando adiciona no código analisado o arquivo selecionado, entretanto o diretório onde os resultados são salvos é alterado para o diretório no qual o arquivo escolhido pelo comando *Compile* está salvo.
2. Comando *Redirect*: Esse comando adiciona no código analisado o arquivo selecionado, entretanto, diferentemente do comando *Compile*, o diretório continua sendo o do arquivo analisado.

Além disso, algumas propriedades como, por exemplo, *mult* presente na classe *LoadShape*, pode exigir uma lista imensa de valores. Por esse motivo, o OpenDSS possibilita a leitura desses valores em arquivos externos que podem estar em formato *.txt ou *.csv. Segue um exemplo:

```
LoadShape.LS1 mult=(File= 'Example.csv')
```

Consulte o Manual de OpenDSS para obter informações adicionais sobre o recurso *File* e as funções *sngFile* e *dblFile*.

5.7 Exemplo

Considere o circuito apresentado na Figura 6.

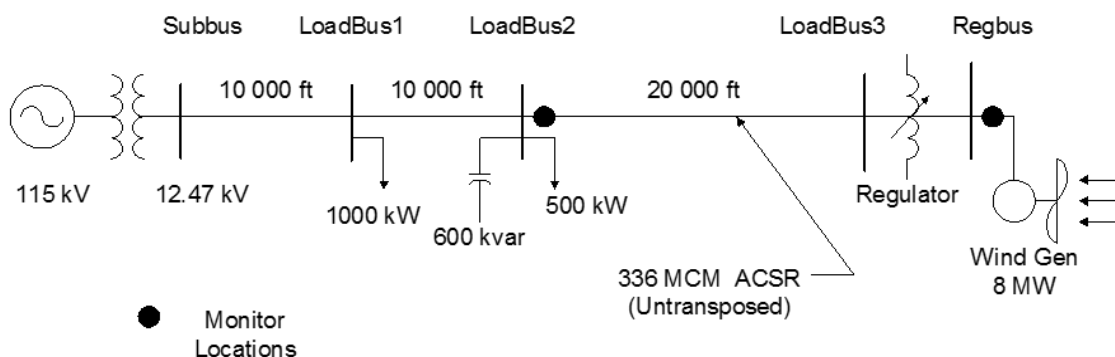


Figura 6: Circuito Exemplo

O código a seguir corresponde a criação e a resolução do circuito da Figura 6 na linguagem de programação do OpenDSS:

```
// Sempre interessante limpar a memoria do OpenDSS
clear
New object=circuit.ExampleCircuit
~ basekv=115 1.00 0.0 60.0 3 20000 21000 4.0 3.0

// Definindo uma curva de carga e uma curva de geracao
New loadshape.day npts=8 interval=3.0
~ mult=(.3 .36 .48 .62 .87 .95 .94 .60)
! Esse exemplo apresenta como utilizar expressoes matematicas com o
  OpenDSS
New loadshape.wind npts=2400 interval={1 24 /} ! As
  unidades sao horas
~ mult=(file=zavwind.csv) action=normalize

// Definindo um arranjo - unbalanced 336 MCM ACSR connection
New linecode.336matrix nphases=3
! Em ohms por 1000 ft
~ rmatrix=(0.0868455 | 0.0298305 0.0887966 | 0.0288883 0.0298305
  0.0868455)
! Em ohms por 1000 ft
~ xmatrix=(0.0205449 | 0.0847210 0.1961452 | 0.0719161 0.0847210
  0.0205449)
! Em ohms por 1000 ft
~ cmatrix=(2.74 | -0.70 2.96 | -0.34 -0.71 2.74)
~ Normamps = 400 Emergamps=600

// Definindo o transformador da subestacao
New transformer.subxfrm phases=3 windings=2 buses=(SourceBus subbus)
~ conns='delta wye' kvs=(115 12.47) kvas=(20000 20000) XHL=7

// Definindo linhas
New line.line1 bus1=subbus bus2=loadbus1 linecode=336matrix length=10
New line.line2 loadbus1 loadbus2 336matrix 10
New line.line3 Loadbus2 loadbus3 336matrix 20

// Definindo cargas
New load.load1 bus1=loadbus1 phases=3 kv=12.47 kw=1000.0 pf=0.88 model
  =1
~ class=1 duty=day
New load.load2 bus1=loadbus2 phases=3 kv=12.47 kw=500.0 pf=0.88 model=1
~ class=1 duty=day conn=delta

// Definindo um capacitor e seu controle
New capacitor.Cap1 bus1=loadbus2 phases=3 kvar=600 kv=12.47
New capcontrol.Cap1Ctrl element=line.line3 terminal=1 capacitor=Cap1
~ type=current ctratio=1 ONsetting=60 OFFsetting=55 delay=2

// Transformador que conecta a GD na rede
```

```
New transformer.reg1 phases=3 windings=2
~ buses=(loadbus3 regbus)
~ conns= 'wye wye'
~ kvs=(12.47 12.47)
~ kvas=(8000 8000)
~ XHL=1

// Definindo os reguladores
New regcontrol.subxfrmCtrl transformer=subxfrm winding=2 vreg=125
~ band=3 ptratio=60 delay=10
New regcontrol.reg1Ctrl transformer=reg1 winding=2 vreg=122 band=3
~ ptratio=60 delay=15

// Definindo a geracao eolica de 8MW
New generator.gen1 bus1=regbus kV=12.47 kW=8000 pf=1 conn=delta
~ duty=wind Model=1

// Definindo monitores para obter as grandezas eletricas escolhidas
! Monitor que armazena a potencia fornecida pela Wind no intervalo de
simulacao
New Monitor.gen1 element=generator.gen1 terminal=1 mode=1
! Monitor que armazena as tensoes e correntes da barra LoadBus2
New Monitor.loadbus2 load.load2 1 mode=0
! Monitor que armazena as tensoes e correntes sequencias no inicio da
linha 3
New Monitor.line3 line.line3 1 mode=48
! Para plotar o perfil de tensao eh necessario um energymeter no inicio
do circuito
New Energymeter.em1 line.line1

// Definindo as tensoes de base do circuito
Set voltagebases=(115 12.47 .48)

// Calculando as tensoes de base para fornecer as respostas , tambem, em
pu
Calcvoltagebases

// Comando que faz o cap e reg controles operarem em sincronismo
// com a simulacao
Set controlmode=time
// Opcoes de simulacao para realizar o estudo em 86400 sec , ou seja , 24
h
// Comecando com em hour 0, second 0
Set mode=duty number=86400 hour=0 stepsize=1 sec=0

// Resolvendo
Solve

// Apresentar os resultados
```

```
! Grafico da tensao na Barra LoadBus2 durante o dia
Plot monitor, object=loadbus2, Channels=(1,3,5)
! Vizualizando a potencia que flui pela linha 1 para o ultimo segundo
    simulado
Visualize element=Line.line1 what=powers
! Grafico do perfil de tensao para o ultimo segundo simulado
Plot profile
```

6 Introdução à COM Interface

Enquanto muito pode ser feito com a versão *Standalone* do OpenDSS, usuários experientes podem abrir um mundo inteiramente novo de aplicações ao aprender a utilizar a interface COM efetivamente. Essa interface permite desenvolver algoritmos em outros programas como, VBA Excel, MatLab, Python, dentre outros, com o objetivo de controlar o OpenDSS e fazer algo que não está implementado atualmente no seu código fonte. Por exemplo, um algoritmo de otimização específica. Apenas alguns estão atualmente implementados no *software*, porém algoritmos externos de grande complexidade podem ser escritos. Esses algoritmos utilizam então o OpenDSS para representar o comportamento do sistema de distribuição, enquanto ajustam quaisquer que sejam as variáveis que se deseja otimizar.

Uma boa capacidade que a interface COM fornece aos usuários é a de usar *loops* nos códigos externos, pois a linguagem de programação do OpenDSS não possui essa capacidade. O comando da linguagem do OpenDSS que mais se assemelha a um *loop* é o comando *Next*, o qual pode simplificar códigos nos quais o tempo é incrementado. Mesmo que houvesse a capacidade de se realizar *loops* na linguagem do OpenDSS, eles seriam executados em uma velocidade relativamente reduzida devido a necessidade de o *loop* ainda ter que ser interpretado. Códigos com *loops* são relativamente fáceis de serem escritos em outras linguagens de programação, onde geralmente são executados com rapidez.

A documentação detalhada da interface COM é fornecida no manual do OpenDSS, no OpenDSS Wiki e nas notas técnicas (veja as referências na Seção 8). Porém, a documentação mais atualizada da interface COM se encontra nela mesma. Utilize um pesquisador de biblioteca de tipos (do inglês, *type object browser* (TLB)) para visualizar as classes, objetos, métodos e interfaces da biblioteca da interface COM do OpenDSS. Um pesquisador muito utilizado é o fornecido no editor do *Visual Basic*, da Microsoft. Para expor a biblioteca da interface COM do OpenDSS, no Excel, abra o editor do *Visual Basic*, acesse Ferramentas → Referências e adicione a referência OpenDSS Engine. Assim, você será capaz de pesquisar a biblioteca da

interface COM do OpenDSS usando o pesquisador de objetos (Exibir → Pesquisador de Objetos) no editor do VBA. Para assistir ao vídeo com uma breve explicação, clique [aqui](#).

Nesse documento, um código em VBA será utilizado como exemplo porque talvez seja a linguagem mais fácil de se entender, independentemente da base que se tem em programação. Além disso, é amplamente disponível. Você pode executar esses exemplos usando o editor de macros do Excel ou na versão livre do *Visual Basic* chamada *Visual Studio Express*. Trechos de códigos em MATLAB (uma linguagem comumente utilizada para computação científica) e Python (uma linguagem de programação popular que tem ganhado uma grande popularidade na comunidade científica) também são fornecidos. Porém, usuários já conseguiram utilizar a biblioteca da interface COM do OpenDSS em uma variedade de linguagens, incluindo C++, C# e R.

6.1 Começando com a interface COM

Em primeiro lugar, certifique-se de que o ambiente de programação que você está usando está conectado à interface COM do OpenDSS. No Visual Basic, isso é feito em Ferramentas → Referências ou em Projetos → Adicionar Referência dependendo de qual versão você está utilizando. Então, selecione a biblioteca de tipos COM OpenDSSEngine. Para assistir ao vídeo com uma breve explicação, clique [aqui](#).

Para instanciar um objeto do OpenDSS e criar um link para as funções comumente utilizadas, use o seguinte código:

```
' Declarando as variaveis como interfaces (classes)
Dim DSSObj As OpenDSSEngine.DSS
Dim DSSText As OpenDSSEngine.Text
Dim DSSCircuit As OpenDSSEngine.Circuit
Dim DSSSolution As OpenDSSEngine.Solution

' Criando o Objeto OpenDSS
DSSObj = New OpenDSSEngine.DSS

' Verificando se o objeto OpenDSS foi criado com sucesso
If Not DSSObj.Start(0) Then
    MsgBox("Unable to start the OpenDSS Engine")
    Return
End If

' Criando atalhos para as 3 principais interfaces
DSSText = DSSObj.Text
```



```
DSSCircuit = DSSObj.ActiveCircuit  
DSSSolution = DSSCircuit.Solution
```

Como pode ser visto, do objeto DSS pai, derivam várias classes filhas. As principais são:

- Interface *Text*
- Interface *Circuit*
- Interface *Solution*

6.2 Interface *Text*

A interface *Text* é a interface mais simples e uma das mais utilizadas. Essa interface permite executar linhas de comandos na linguagem do OpenDSS, ou seja, qualquer comando da versão *Standalone* está imediatamente disponível para o usuário.

Para usar essa interface, defina a sua propriedade *Command* com um comando da versão *Standalone*, conforme apresentado a seguir:

```
' Carregando o arquivo IEEE123Master.dss  
DSSText.Command = "Compile 'C:\example\IEEE123Master.dss'"  
' Criando um novo capacitor atraves da interface Text  
DSSText.Command = "New Capacitor.C1 Bus1=1 Phases=3 kVAR=1200"  
DSSText.Command = "~ Enabled=false"  
' Alterando a barra 1 da linha L1  
DSSText.Command = "Line.L1.Bus1 = 5"
```

Se o comando utilizado gera algum relatório, o resultado pode ser recuperado usando a propriedade *Result* da interface *text*, conforme exemplo a seguir:

```
' Exportar as tensoes para uma arquivo *.csv  
Dim Filename As String  
DSSText.Command = "Export Voltages"  
Filename = DSSText.Result  
MsgBox("File saved to: " & Filename)
```

6.3 Interface *Circuit*

A interface *Circuit* é comumente utilizada para editar as propriedades dos vários elementos em um circuito. Quase todas as classes de elementos do OpenDSS como Linhas, Cargas e Capacitores, possuem uma classe filho sob a interface *Circuit*.

Essas classes filho também possuem funções convenientes que permitem a iteração ao longo dos elementos membros, que pode ser útil quando se utiliza um *loop* no algoritmo externo. Por exemplo, no código a seguir é utilizado um *loop* com objetivo de incrementar em 20% a potência ativa de todas as cargas.

```
' Passe por todas as cargas e incremente a potencia
Dim iLoads As Integer

iLoads = DSSCircuit.Loads.First 'Configurar a primeira carga da lista
    de cargas como ativa
While iLoads
    ' Multiplicar a potencia ativa da carga selecionada por 1.2
    DSSCircuit.Loads.kW = DSSCircuit.Loads.kW * 1.2
    ' Passe para a proxima carga da lista , tornando-a a carga ativa
    iLoads = DSSCircuit.Loads.Next
End While
```

Caso seja necessária a edição de um elemento em particular, deve-se utilizar o método *SetActiveElement* e a interface *ActiveDSSElement*, conforme apresentado no exemplo a seguir:

```
' Defina a potencia nominal do capacitor C83 (kvar) para 1200
DSSCircuit.SetActiveElement("Capacitor.C83")
DSSCircuit.ActiveDSSElement.Properties("kVAR").Val = 1200
```

Entretanto, a vantagem deste método sobre simplesmente usar a interface *Text*, como mostrado abaixo, é discutível. Use o método que lhe seja mais conveniente.

```
' Defina a potencia nominal do capacitor C83 (kvar) para 1200
' Atraves da Interface Text
DSSText.Command = "Capacitor.C83.kVAR = 1200"
```

Finalmente, outra boa característica da interface *Circuit* é a possibilidade de obter resultados tais como tensões nas barras, perdas nos elementos, entre outros. No exemplo abaixo são obtidos os nomes e as tensões das barras:

```
' Obter o nome das barras e as tensoes nodais
Dim BusNames As String()
Dim Voltages As Double()
BusNames = DSSCircuit.AllBusNames
Voltages = DSSCircuit.AllBusVmagPu

' Mostre em um mensagem de box qual e a tensao na quinta barra da lista
MsgBox(BusNames(5) & " voltage mag in per unit is: " & Voltages(5))
```

Para mais informações sobre como obter resultados do fluxo de potência a partir da interface *Circuit*, veja em “All commands” (*AllElementLoses*, *AllNodeVmagByPhase*, etc...) na seção COM no manual do OpenDSS.

6.4 Interface *Solution*

A Interface *Solution* é usada para monitorar e controlar o processo de solução e o procedimento de controle do OpenDSS. Isso inclui resolver o circuito, definir o modo de solução, monitor a convergência e outros aspectos da resolução de um circuito no OpenDSS.

O código a seguir apresenta a solução do circuito e a verificação de sua convergência através da interface *Solution*:

```
' Resolva o circuito
DSSSolution.Solve()
If DSSSolution.Converged Then
MsgBox("O circuito foi resolvido com sucesso")
End If
```

Também é possível controlar a solução do OpenDSS minuciosamente. Por exemplo, digamos que se queira modelar os efeitos da inserção de cargas após 30 segundos de simulação. O código abaixo pode ser utilizado para tal:

```
'Modelagem do efeito da insercao de cargas no sistema
DSSText.Command = "New Monitor.Mon1 element=Line.L100 mode=0"
DSSSolution.StepSize = 1 ' Configura o passo de tempo para 1 segundo
DSSSolution.Number = 30 ' Resolve 30 segundos de simulacao

' Configura o mode de simulacao para duty cycle , o qual forca as cargas
a usar
' a curva de carga 'duty' e permite uma simulacao temporal
DSSSolution.Mode = OpenDSSengine.SolveModes.dssDutyCycle
DSSSolution.Solve()

DSSCircuit.Enable("Load.L1") 'Ativa as cargas
DSSSolution.Number = 30 'Resolve mais 30 segundos de simulacao
DSSSolution.Solve()

MsgBox("Seconds Elapsed:" & DSSSolution.Seconds)
'Plota a tensao nos 60 segundos de simulacao
DSSText.Command = "Plot Monitor object=Mon1 Channels=(1,3,5)"
```

Por fim, a interface circuit fornece informações muito específicas quanto ao método e controle por trás da solução. Usando os vários métodos de solução(por exemplo,

Solve, *SolveDirect*, *SolveNoControl*, etc...) e outras funções como *SystemYChanged* and *MostIterationsDone*, um controle minucioso e informações do processo de solução podem ser obtidos. Utilizando a função *CheckControls* e outros métodos da interface *Solution* junto com a interface *CtrlQueue*, esquemas de controle inteligentes podem ser implementados. Os detalhes dessas interfaces está além do escopo desse documento. Mais informações podem ser encontradas nas notas técnicas do OpenDSS (veja as referências na Seção 8).

7 Controlando o OpenDSS via COM

Nessa seção são apresentados exemplos de controle do OpenDSS via COM utilizando os seguintes *softwares*:

1. VBA Excel
2. MATLAB
3. Python

7.1 Exemplo de Controle Utilizando o VBA Excel

O código apresentado a seguir é idêntico aos trechos apresentados ao longo desse documento, entretanto de forma unificada. Clique [aqui](#) para ser direcionado a uma série de vídeos que apresenta exemplos de como controlar o OpenDSS utilizando o VBA Excel.

```
' Declarando as variaveis como interfaces (classes)
Dim DSSObj As OpenDSSEngine.DSS
Dim DSSText As OpenDSSEngine.Text
Dim DSSCircuit As OpenDSSEngine.Circuit
Dim DSSSolution As OpenDSSEngine.Solution

' Criando o Objeto OpenDSS
DSSObj = New OpenDSSEngine.DSS

' Verificando se o objeto OpenDSS foi criado com sucesso
If Not DSSObj.Start(0) Then
    MsgBox("Unable to start the OpenDSS Engine")
    Return
End If

' Criando atalhos para as 3 principais interfaces
DSSText = DSSObj.Text
```

```

DSSCircuit = DSSObj.ActiveCircuit
DSSSolution = DSSCircuit.Solution

' Carregando o arquivo IEEE123Master.dss
DSSText.Command = "Compile 'C:\example\IEEE123Master.dss'"
' Criando um novo capacitor atraves da interface Text
DSSText.Command = "New Capacitor.C1 Bus1=1 Phases=3 kVAR=1200"
DSSText.Command = "~ Enabled=false"
' Alterando a barra 1 da linha L1
DSSText.Command = "Line.L1.Bus1 = 5"

' Exportando tensoes em csv file
Dim Filename As String
DSSText.Command = "Export Voltages"
Filename = DSSText.Result
MsgBox("File saved to: " & Filename)

' Passe por todas as cargas
Dim iLoads As Integer ' Verificar qual cargas esta selecionada

iLoads = DSSCircuit.Loads.First
While iLoads
    ' Multiplicar a potencia ativa da carga selecionada por 1.2
    DSSCircuit.Loads.kW = DSSCircuit.Loads.kW * 1.2
    ' Passe para a proxima carga da lista
    iLoads = DSSCircuit.Loads.Next
End While

' Defina a potencia nominal do capacitor C83 (kvar) para 1200
DSSCircuit.SetActiveElement("Capacitor.C83")
DSSCircuit.ActiveDSSElement.Properties("kVAR").Val = 1200

' Defina a potencia nominal do capacitor C83 (kvar) para 1200
' Atraves da Interface Text
DSSText.Command = "Capacitor.C83.kVAR = 1200"

' Armazenar as tensoes nodais
Dim BusNames As String()
Dim Voltages As Double()
BusNames = DSSCircuit.AllBusNames
Voltages = DSSCircuit.AllBusVmagPu

' Mostre em mensagem de box qual eh a tensao na quinta barra da lista
MsgBox(BusNames(5) & " voltage mag in per unit is: " & Voltages(5))

' Resolva o circuito
DSSSolution.Solve()
If DSSSolution.Converged Then
    MsgBox("O circuito foi resolvido com sucesso")

```

End If

7.2 Exemplo de Controle Utilizando o MATLAB

O código em MATLAB mostrado abaixo executa as mesmas funções que os trechos de código apresentados nesse documento. Clique [aqui](#) para ser direcionado a uma série de vídeos que apresenta exemplos de como controlar o OpenDSS utilizando o MATLAB.

```
clc
clear all
close all

% Criando o Objeto OpenDSS
DSSObj = actxserver('OpenDSSEngine.DSS');

% Verificando se o objeto OpenDSS foi criado com sucesso
if ~DSSObj.Start(0),
    disp('Unable to start the OpenDSS Engine')
    return
end

% Criando atalhos para as 3 principais interfaces
DSSText = DSSObj.Text;
DSSCircuit = DSSObj.ActiveCircuit;
DSSSolution = DSSCircuit.Solution;

% Carregando o arquivo IEEE123Master.dss
DSSText.Command = 'Compile "C:\example\IEEE123Master.dss"';

% Criando um novo capacitor atraves da interface Text
DSSText.Command = 'New Capacitor.C1 Bus1=1 Phases=3 kVAR=1200';
DSSText.Command = '~ Enabled=false';

% Alterando a barra 1 da linha L1
DSSText.Command = 'Line.L1.Bus1 = 5';

% Exportando tensoes em csv file
DSSText.Command = 'Export Voltages';
Filename = DSSText.Result;
disp(['File saved to: ' Filename])

% Passe por todas as cargas
iLoads = DSSCircuit.Loads.First;
while iLoads,
    % Multiplicar a potencia ativa da carga selecionada por 1,2
```

```
DSSCircuit.Loads.kW = DSSCircuit.Loads.kW * 1.2;
    % Passe para a proxima carga da lista
    iLoads = DSSCircuit.Loads.Next;
end

% Defina a potencia nominal do capacitor C83 (kvar) para 1200
DSSCircuit.SetActiveElement('Capacitor.C83');
DSSCircuit.ActiveDSSElement.Properties('kVAR').val = '1200';

% Armazenar as tensoes nodais
BusNames = DSSCircuit.AllBusNames;
Voltages = DSSCircuit.AllBusVmagPu;

% Mostre em mensagem de box qual eh a tensao na quinta barra da lista
disp([BusNames{5} ' voltage mag in per unit is: ' num2str(Voltages(5))
])

% Resolva o circuito
DSSSolution.Solve();
if DSSSolution.Converged,
    disp('The Circuit Solved Successfully')
end
```

7.3 Exemplo de Controle Utilizando o Python

O código em Python mostrado abaixo executa as mesmas funções que os trechos de código apresentados nesse documento. Clique [aqui](#) para ser direcionado a uma série de vídeos que apresenta exemplos de como controlar o OpenDSS utilizando o Python.

```
import win32com.client

# Criando o Objeto OpenDSS
try:
    DSSObj = win32com.client.Dispatch("OpenDSSEngine.DSS")
except:
    print "Unable to start the OpenDSS Engine"
    raise SystemExit

# Criando atalhos para as 3 principais interfaces
DSSText = DSSObj.Text
DSSCircuit = DSSObj.ActiveCircuit
DSSSolution = DSSCircuit.Solution

# Carregando o arquivo IEEE123Master.dss
```

```
DSSText.Command = r"Compile 'C:\example\IEEE123Master.dss'"
# Criando um novo capacitor atraves da interface Text
DSSText.Command = "New Capacitor.C1 Bus1=1 Phases=3 kVAR=1200"
DSSText.Command = "~ Enabled=false"
# Alterando a barra 1 da linha L1
DSSText.Command = "Line.L1.Bus1 = 5"

# Exportando tensoes em csv file
DSSText.Command = "Export Voltages"
Filename = DSSText.Result
print "File saved to: " + Filename

# Passe por todas as cargas
iLoads = DSSCircuit.Loads.First
while iLoads:
    # Multiplicar a potencia ativa da carga selecionada por 1,2
    DSSCircuit.Loads.kW = DSSCircuit.Loads.kW * 1.2
    # Passe para a proxima carga da lista
    iLoads = DSSCircuit.Loads.Next

# Defina a potencia nominal do capacitor C83 (kvar) para 1200
DSSCircuit.SetActiveElement("Capacitor.C83")
DSSCircuit.ActiveDSSElement.Properties("kVAR").Val = 1200

# Armazenar as tensoes nodais
BusNames = DSSCircuit.AllBusNames
Voltages = DSSCircuit.AllBusVmagPu

# Mostre em mensagem de box qual eh a tensao na quinta barra da lista
print BusNames[5] + " voltage mag in per unit is: " + str(Voltages[5])

# Resolva o circuito
DSSSolution.Solve()
if DSSSolution.Converged:
    print "The Circuit Solved Successfully"
```

8 Recursos Adicionais

Ao continuar a usar o OpenDSS e explorar a sua capacidade, você deverá se deparar com situações que requerem mais informações do que as motradas nesse guia introdutório. Por esse motivo, os desenvolvedores do OpenDSS e a comunidade têm fornecido vários recursos para os usuários.

8.1 Onde ir a partir daqui?

Com as informações nesse documento, você deve estar preparado para começar a utilizar o *software*. Uma extensa série de exemplos de recursos do OpenDSS são fornecidos no diretório **Examples** na pasta onde o OpenDSS foi instalado. Para assistir um vídeo com uma breve explicação sobre onde encontrar esses arquivos, clique [aqui](#). Os principais recursos são:

1. Exemplos de códigos no diretório **Example/Scripts**.
2. Exemplos de simulações em diferentes modos: sequenciais no tempo; análise de Monte Carlo; análise dinâmica; análise harmônica; estudo de curto-circuito; otimização da localização de capacitores.
3. Exemplos da utilização de vários elementos, tais como:
 - Reguladores de tensão e os seus controles, capacitores e o seus controles, e controladores de otimização volt-VAR centralizados.
 - Geradores, sistemas fotovoltaicos, geradores com fator de potência fixo, geradores de indução, baterias, dentre outros.
 - Fusíveis, religadores, relês e elementos de curto-circuito.
 - Curvas de carga, curvas de geração, curvas de crescimento de demanda, espectro harmônicos, arranjos de cabos, curvas de temperatura, medidores e monitores.
4. Exemplos do controle do OpenDSS utilizando diferentes *software*.
5. Modelos de alimentadores reais são fornecidos no diretório **EPRITestCircuits**.
6. Um modelo de uma linha de transmissão simples é fornecido no diretório **Examples/Stevenson**.
7. No diretório **Paulo_Example** está o trabalho de conclusão de curso entregue à Escola Politécnica da Universidade de São Paulo realizado por Paulo Radatz, em português.
8. No diretório **Tranning** encontram-se alguns arquivos *PowerPoint* realizados por Roger Dugan.

8.2 Recursos de Referência

O OpenDSS fornece vários recursos de referência para documentar extensivamente as diversas características da linguagem de programação do OpenDSS e da interface COM.

- O botão de **Ajuda** localizado no *software*, que fornece uma referência rápida às diversas propriedades dos elementos e aos comandos do OpenDSS.
- O **manual do OpenDSS**, localizado em **Doc/OpenDSSManual.pdf** no diretório de instalação, que fornece informações muito mais aprofundadas do que as apresentadas nesse manual introdutório.
- As **notas técnicas**, localizadas no diretório **Docs** na local de instalação do OpenDSS, que são uma série de relatórios que fornecem informações específicas e detalhadas da interface COM, atributos do OpenDSS e outras aplicações do programa tais como a modelagem dos efeitos do núcleo de transformadores e sistemas fotovoltaicos.
- O **OpenDSS Wiki**, localizado [aqui](#), que fornece informações que também podem ser encontradas nas notas técnicas e no manual do OpenDSS.
- Um **pesquisador de biblioteca de tipos** (do inglês, *type object browser* (TLB)) para visualizar as classes, objetos, métodos e interfaces da biblioteca da interface COM do OpenDSS. Um pesquisador muito utilizado é o fornecido no editor do *Visual Basic*, da Microsoft. Para expor a biblioteca da interface COM do OpenDSS, no Excel, abra o editor do *Visual Basic*, acesse Ferramentas → Referências e adicione a referência OpenDSS Engine. Assim, você será capaz de pesquisar a biblioteca da interface COM do OpenDSS usando o pesquisador de objetos (Exibir → Pesquisador de Objetos) no editor do VBA. Para assistir ao vídeo com uma breve explicação, clique [aqui](#).

8.3 Ajuda Adicional

Se você precisar de ajuda adicional, os fóruns de OpenDSS no SourceForge fomentam uma próspera comunidade que tem como objetivo sanar as suas dúvidas. Clique [aqui](#) para ser direcionado aos fóruns.

8.4 Contato

Caso você tenha encontrado algum erro nesse documento, por favor, contate os autores/tradutores no seguintes e-mails:

paulo.radatz@gmail.com
celsorocha11@gmail.com

Obrigado!