

Fernando Luis Spolador

Software interativo com acesso a bancos de dados, modelos de componentes e métodos numéricos visando estudos para a geração de cenários de operação de redes de transmissão de energia

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Engenharia Elétrica.

Orientador: Prof. Dr. Anésio dos Santos Júnior.

Campinas, SP

2008

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Sp65s Spolador, Fernando Luis
Software interativo com acesso a banco de dados,
modelos de componentes e métodos numéricos visando
estudos para a geração de cenários de operação de redes de
transmissão de energia
Fernando Luis Spolador. – Campinas, SP:
[s.n.], 2008.

Orientadore: Anésio dos Santos Júnior.
Dissertação (Mestrado) - Universidade Estadual de Campinas,
Faculdade de Engenharia Elétrica e de Computação.

1. . Energia elétrica - Transmissão. 2. Interfaces gráficas
de usuários (Sistema de computador). 3. Programação
orientada a objetos (Computação). I. Santos Júnior, Anésio.
II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação. III.
Título

Título em Inglês: Interactive software with access to database, components models and
numerical methods for generating of operation scenarios for energy
transmission system
Palavras-chave em Inglês: Power system, Object oriented, Graphical user interface
Área de concentração: Energia Elétrica
Titulação: Mestre em Engenharia Elétrica
Banca Examinadora: Luiz Carlos Pereira da Silva, Paulo Sérgio da Silva
Data da defesa: 27/06/2008

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Fernando Luis Spolador

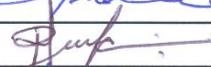
Data da Defesa: 27 de junho de 2008

Título da Tese: "Software Interativo com Acesso a Bancos de Dados, Modelos de Componentes e Métodos Numéricos Visando Estudos para a Geração de Cenários de Operação de Redes de Transmissão de Energia"

Prof. Dr. Anésio dos Santos Júnior (Presidente):



Prof. Dr. Paulo Sérgio da Silva:



Prof. Dr. Luiz Carlos Pereira da Silva:



Resumo

Nesta dissertação é apresentada a proposta de um software para simulação e análise de sistemas de potência. A crescente necessidade de expansão dos sistemas elétricos e de validação destes sistemas nos mais diversos estados, fazem com que sejam necessários cada vez mais estudos. As simulações computacionais se apresentam como uma forma eficaz e de custo reduzido para a realização destes estudos, porém estas simulações acabam por gerar uma enorme quantidade de informações. O software proposto possui um conjunto de recursos, dentre eles, interfaces gráficas amigáveis com o usuário e manipulação de base de dados, que buscam fornecer as ferramentas necessárias para o processo de análise e efetiva busca de soluções para o modelo do sistema elétrico estudado. O software foi estruturado de forma a facilitar uma evolução contínua e a inserção de novos recursos e funcionalidades.

Palavras-chave: Sistema de Potência, Orientação a Objetos, Interface Gráfica com o Usuário.

Abstract

This dissertation presents the proposal for a software for simulation and analysis of power systems. The growing need for expansion of electrical systems, the need for validation of these systems in the most diverse state, make it increasingly necessary studies of these systems. The computer simulations are presented as an effective and low cost for carrying out this study, but these simulations ultimately generate a huge amount of information. The software proposed has a set of resources, including friendly graphical user interfaces and manipulation of the databases, which seek to provide the tools necessary to the process of analysis and effective search for solutions to the model of the electrical system studied. The software has been structured to facilitate an ongoing evolution and inclusion of new features and functionality.

Keywords: Power System, Object Oriented, Graphical User Interface.

Dedicatória

Agradeço ao Prof. Anésio, pela sua paciência e orientação.

A Gabriela pelo fornecimento da rotina de cálculo que usamos.

Aos colegas do DENSIS, em especial José, Ana Claudia, Iara, André, Gabriela e Anibal pelos momentos de diversão e descontração.

A minha família, que mesmo de longe, sempre esteve presente.

À CAPES, pelo apoio financeiro.

Aos meus pais, ao meu irmão e a minha irmã

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
1 Introdução	1
2 Justificativa	3
2.1 Visualização Científica	4
2.2 Aplicabilidade	4
2.3 Sistemas Existentes	6
2.4 Software Proposto	9
2.5 Conclusão	10
3 Formulação do Problema através de Cálculos Seqüenciais de Fluxo de Potência e a Estratégia do Software Interativo Proposto	11
3.1 Equações do Fluxo de Potência	12
3.2 Restrições operacionais e limites a serem atendidos	15
3.3 Estratégia do Software Interativo	18
3.4 Conclusão	21
4 Projeto do sistema	22
4.1 Conceitos de Orientação a Objetos	23
4.2 Recursos Computacionais Utilizados	24
4.3 Visão do Usuário	27
4.4 Diagramas de Projeto	29
4.4.1 Base de Dados	30

4.4.2	Controle de Dados	32
4.4.3	Construção de Gráficos	34
4.4.4	Elementos do Sistema Elétrico	35
4.4.5	Interface gráfica	37
4.4.6	Controle de módulos	37
4.5	Conclusão	39
5	Descrição das funcionalidades do sistema proposto	40
5.1	Agrupamento de dados para acesso às funcionalidades do sistema	41
5.1.1	Visualização do modelo	44
5.1.2	Visualização dos gráficos	51
5.1.3	Visualização dos dados sistema	55
5.1.4	Controle da simulação	56
5.2	Navegação autônoma pelo sistema	58
5.3	Inserção de novos módulos de cálculo	62
5.3.1	Interface para a inserção de novos módulos	67
5.4	Conclusão	68
6	Conclusão e trabalhos futuros	70
6.1	Trabalhos Futuros	71
	Referências bibliográficas	72

Lista de Figuras

3.1	Figura do caso de teste IEEE de 14 barras, obtido de [IEE]	13
3.2	Fluxo de potência ativa e reativa em um ramo	15
3.3	Descrição da Estratégia do Interação	20
4.1	Visão do usuários das funcionalidades do sistema	27
4.2	Diagrama de pacotes	29
4.3	Diagrama de classes, visão banco de dados	30
4.4	Modelo banco de dados	31
4.5	Diagrama de classes, visão controle de dados	32
4.6	Diagrama de classes, visão gráficos	35
4.7	Diagrama de classes, visão elementos do Sistema Elétrico	36
4.8	Diagrama de classes, visão controle de módulos	38
5.1	Interface para Agrupamento de Dados do Modelo	42
5.2	Interface para Agrupamento de Dados do Modelo, com grupos destacados	44
5.3	Interface para Exibição de Barras Vizinhas	45
5.4	Interface para Configuração de Campos	46
5.5	Interface para Configuração de Cores	47
5.6	Interface para Configuração de Limites	48
5.7	Interface com Exibição de Gráficos juntos às Barras	49
5.8	Interface para Controle de Contingências	50
5.9	Interface para Inserção de Novos Elementos	51
5.10	Interface para Exibição de Gráficos	52
5.11	Interface com Funcionalidades para Gráficos	53
5.12	Interface para Exibição de Gráficos em Círculos	54

5.13 Interface para Exibição de Potência Reativa	55
5.14 Interface para Exibição Tabular dos Dados	56
5.15 Interface para Controle de Simulação	57
5.16 Interface Principal do Software	59
5.17 Resultado da Abordagem SDI	60
5.18 Interface Principal para Controle de Simulação	61
5.19 Interface para Inserção de Módulos de Cálculo	68

Lista de Tabelas

5.1	Métodos da Classe AlgoritmoBaseFP para obtenção de informações do modelo . . .	63
5.2	Métodos da Classe AlgoritmoBaseFP para salvar dados das barras obtidos nas simulações	64
5.3	Métodos da Classe AlgoritmoBaseFP para salvar dados dos ramos e transformadores obtidos na simulação	65
5.4	Métodos da Classe AlgoritmoBase	66
5.5	Constantes para retorno de valores de erro	67

Capítulo 1

Introdução

Em todo o mundo, o progresso industrial e o aumento da qualidade de vida de uma população passam pelo desenvolvimento de fontes de energia. Nos dias atuais ainda é importante que o fornecimento dessa energia seja feito com o menor impacto possível ao meio-ambiente. Os sistemas elétricos de potência são uma das importantes ferramentas para se alcançar estes objetivos. Eles são compostos, basicamente, por: conversores de energia, redes de transmissão e sistemas de distribuição [SJ82].

As decisões sobre o despacho de potência para os geradores de energia devem ser orientadas de forma a minimizar o custo de produção e o impacto ambiental, sendo que junto ao custo de geração há o custo de transmissão dessa energia até os sistemas de distribuição.

O estudo para a validação de estratégias de operação do sistema em estado normal ou em estados anormais [GSJ94], os estudos para levantamento das necessidades de expansão, entre outros, fazem com que sejam necessárias ferramentas para auxiliar engenheiros de potência e operadores em suas decisões técnicas. É objetivo desta dissertação apresentar uma proposta de sistema que visa facilitar diversos aspectos deste processo de planejamento da operação e da expansão dos sistemas elétricos de transmissão.

O software proposto possui um conjunto de interfaces amigáveis com o usuário, é capaz de gerenciar uma base de dados formada pelos modelos das redes elétricas e pelos resultados das simulações e também possui a capacidade de receber novos módulos de cálculo para o fluxo de potência. Com estes recursos procura-se criar uma ferramenta de auxílio a operadores e engenheiros

de potência.

Esta dissertação está organizada da seguinte maneira:

- No Capítulo dois são abordadas algumas propostas presentes na literatura para sistemas de mesmas natureza do aqui proposto, assim como alguns software comerciais disponíveis. Também é feita uma revisão das possibilidades gerais de aplicação desses sistemas na área de potência.
- No Capítulo três são descrito o problema do fluxo de potência, as estratégia usada para se buscar soluções e como o sistema desenvolvido fornece ferramentas para se alcançar os resultados necessários.
- No Capítulo quatro é feita uma descrição dos módulos do sistema desenvolvido neste trabalho, apresentando a engenharia de software utilizada para construir um protótipo do sistema proposto.
- No Capítulo cinco são apresentadas as funcionalides presentes no propótipo desenvolvido, mostrando os recursos disponíveis para análise, alterações e simulações dos sistemas de potência.
- No Capítulo seis são apresentadas as conclusões e propostas para futuros desenvolvimentos, dando continuidade à evolução do sistema aqui proposto.

Capítulo 2

Justificativa

A enorme quantidade de dados e informações fornecidas por sistemas de computação científica, despertam, desde o fim da década de 80 [LQ98], a necessidade de construção de meios efetivos de visualização e análise destes dados. Estes meios devem permitir a pesquisadores e engenheiros trabalharem de forma mais efetiva com os resultados obtidos das simulações de seus objetos de estudo.

Em diversas áreas como meteorologia, biologia molecular, simulação militar e física, por exemplo, são desenvolvidas ferramentas com o intuito de facilitar a manipulação de dados científicos [SG93]. Em sistemas de potência, porém, ainda há uma vasta gama de possibilidades a serem exploradas, visto que, em muitos casos, os dados ainda são arranjados de forma tabular para serem só então analisados [SO04] [TKM⁺00] [KML03].

O acesso aos resultados das simulações de forma tabular acaba por sobrecarregar o usuário do sistema com uma quantidade enorme de informações, obrigando-o, muitas vezes, a tomar decisões baseadas em uma pequena fração dos dados. Decisões tomadas a partir desse tipo de abordagem estão muito mais sujeitas a erros, podendo conduzir a uma avaliação incorreta do sistema de potência [BPW99].

Este capítulo está dividido em quatro seções, a primeira apresenta alguns aspectos de visualização científica, a segunda trata das possíveis aplicações que permitem modelagem, simulações e análise de sistemas de potência. A terceira seção descreve algumas ferramentas proposta na literatura e alguns produtos desenvolvidos pela iniciativa privada. A quarta seção descreve os

aspectos gerais do sistema proposto.

2.1 Visualização Científica

É importante que os sistemas de suporte a investigações científicas garantam que os aspectos mais críticos e de maior exigência cognitiva da área alvo sejam representados de forma adequada. Para isso é necessário compreender as razões de determinadas tarefas apresentarem níveis de dificuldades inerentes a sua especificidade [SG93]. A representação adequada pode ser obtida através da definição de conjuntos de interfaces com o usuário que se mostrem eficientes.

Alguns aspectos em que a visualização científica auxilia as engenharias, de uma forma geral, são: a diminuição da necessidade de testes físicos, visto que quando se faz uma simulação se procura reproduzir algo do mundo real e uma melhor integração entre análise e *design*. Quanto melhor se visualizar os resultados de simulações, mais fácil fica realizar as modificações que podem ser necessárias. Com o avanço dos sistemas de computação fica cada vez mais fácil, rápido e barato realizar simulações. Dessa forma, a visualização científica também contribui para o aumento da efetividade e da precisão das análises [Gal94].

2.2 Aplicabilidade

Dentre as várias aplicabilidades para softwares que possibilitem a visualização científica nos estudos de sistemas de transmissão, pode-se começar destacando a análise dos perfis de tensão e do suporte de potência reativa da rede. Dentre as mais diversas configurações de carga, geração e topologia da rede é extremamente importante o cálculo dos fluxos de potência que cada circuito esta transmitindo.

A análise de contingências em sistemas de potência, por exemplo, é uma área extremamente beneficiada pelo acesso facilitado aos dados das simulações por parte dos engenheiros e pesquisadores, por compreender um tópico crítico e de grande importância na grande maioria das redes de transmissão [SO04] [XLKL99] [LJC⁺88].

Esta importância advém do aumento de complexidade que ocorre tanto nas caracterís-

ticas das cargas quanto nas tecnologias de controle do sistema de suprimento de energia [KML03] [TKM⁺00]. Como consequência, surge a necessidade de ampliação das redes e dos respectivos modelos, fazendo com que seja cada vez mais importante verificar e validar o estado do sistema nas mais diversas condições. Essas análises de contingências acarretam em enorme quantidade de dados, que, sem a devida interpretação, podem não revelar informações essenciais de segurança [Hau99]. Uma das mais importantes tarefas dos engenheiros de potência é manter a estabilidade dinâmica, de tensão, entre outros parâmetros, sob uma grande variedade de restrições [TKM⁺00].

A topologia em que as barras e circuitos estão configurados acaba por definir a característica de funcionalidade com relação aos carregamentos dos circuitos e o perfil de tensão da rede. Pelos motivos citados anteriormente, novas configurações são feitas com uma frequência cada vez maior, podendo consumir muito tempo dos planejadores e operadores do sistemas. Assim, torna-se interessante que as configurações das topologias, das barras e dos circuitos, possam ser feitas de maneira intuitiva por parte dos especialistas [TKM⁺00].

De uma maneira geral, com o passar do tempo, vários sistemas computacionais foram projetados para monitorar e controlar sistemas de potência, inclusive em tempo real. O problema é que grande parte destes sistemas, devido à metodologia de desenvolvimento essencialmente orientada a funções, acabam por se tornarem um grande desafio em termos de manutenção e evolução de software, abrindo espaços para novos desenvolvimentos, baseados em metodologias mais atuais [KML03].

Outro ponto importante é a interface com o usuário, pois além de ajudar pesquisadores e engenheiros a trabalharem com os dados, uma interface gráfica amigável facilita a interação do usuário com o sistema. Dessa forma acaba por auxiliar o acesso aos fluxos de potência ativa e reativa, dentre outros parâmetros e elementos dos sistemas de potência [SLI99].

Outra aplicação para sistemas de software com as características discutidas aqui é no ensino e treinamento de planejadores e operadores de sistemas de potência. Dessa maneira, cria-se a oportunidade para que estudantes, planejadores e operadores percebam e entendam, de forma intuitiva, o funcionamento dos sistemas de potência, possibilitando o aprendizado do conteúdo transmitido em sala de aula [Ove03]. Um sistema capaz de representar dados de simulações em redes complexas pode também fazer representações análogas para redes mais simples, funcionando dessa forma como uma ferramenta de ensino [XLKL99] [SLI99] [LJC⁺88].

2.3 Sistemas Existentes

Nesta seção são discutidas algumas propostas existentes na literatura para a construção de sistemas que auxiliam de alguma forma as tarefas de análise, construção, ampliação, monitoramento e simulação em sistemas elétricos de potência. Também serão discutidos alguns dos principais softwares comerciais.

É apresentado em [HJAA98] um sistema denominado *GINA*¹ que foi desenvolvido com os seguintes intuitos: permitir a construção de diagramas para representar os sistemas elétricos, permitir a verificação de problemas causados por problemas de topologia e permitir a apresentação dos resultados das simulações de forma gráfica ou tabular. O sistema foi projetado de maneira modular para facilitar o seu desenvolvimento, que teve início nos anos 80.

Em [BPW99] é proposto um sistema focado na gerência de redes de subtransmissão de energia. O sistema é composto por três partes principais: um primeiro módulo para exibição dos modelos onde, assim como no segundo módulo de análise, os dados são organizados de forma matricial para facilitar a leitura e um terceiro módulo de verificação dos dados.

Foi proposto em [SLI99] um sistema para o treinamento e ensino, útil para modelagem e análise de sistema de potência. O sistema procura combinar, em uma única interface amigável com o usuário, recursos para modelagem, análise e diversas simulações em sistemas de potência. Foi desenvolvido um editor gráfico, para os modelos unifilares, e um sistema de banco de dados. Esse sistema oferece controle de simulações e animações para apresentar os resultados obtidos. Há um conjunto de rotinas de cálculo pré-definidas. O sistema possui limitações quanto ao tamanho dos casos que pode tratar, as limitações são de cerca de 30 barras e 50 linhas de transmissão.

Em [XLKL99] é proposto o uso de planilhas eletrônicas para a análise de sistemas de potência. É ressaltado que muitos engenheiros de potência já usam planilhas e que, hoje, através de *plugins* muitas simulações podem ser realizadas através destes softwares. Os autores ressaltam as vantagens que o uso deste tipo de software oferece, uma vez que ao realizar os estudos por deles, além das funcionalidades específicas fornecidas pelos *plugins*, todos os outros recursos presentes de forma padrão nas planilhas eletrônicas também podem ser usados. É ressaltado também que, como ferramenta de ensino, este tipo de abordagem permite que os alunos, antecipadamente a

¹Do Inglês: *Graphics Aided Interactive Network Analysis*

aula, obtenham os *templates* das planilhas que serão usadas. Mesmo sem os *plugins* que permitem a análise e a simulação, as planilhas são úteis para que os alunos interajam com o modelo, pois eles podem utilizar os recursos das próprias planilhas.

As desvantagens apresentadas pelos próprios autores, em [XLKL99], para este tipo de abordagem é que as planilhas devem ser feitas para topologias específicas de rede, acarretando em perda de generalidade. O sistema não é adequado para problemas de grande porte e as tarefas numéricas são feitas por rotinas totalmente fechadas e de propósito genérico.

Em [Lav00] é proposto um conjunto de interfaces com o usuário que procuram criar um sistema capaz de ser operado mesmo por usuários não treinados. O sistema é desenvolvido de acordo com alguns princípios básicos como: identificação e caracterização do usuário, identificação e análise das tarefas do usuário, o uso de menus, botões e barras de ferramentas [Lav00].

É proposto em [TKM⁺00] um conjunto integrado de ferramentas para a análise de sistemas de potência, neste trabalho os autores procuram criar um conjunto de interfaces com o usuário que permita, inclusive, a manipulação dos sistemas por pessoas que não são familiarizadas com os cálculos envolvidos em sistemas de potência. O sistema proposto é denominado *Impact*² e oferece duas principais funcionalidades: uma interface gráfica com o usuário para a inserção de dados e visualização dos resultados das simulações, a segunda funcionalidade diz respeito à capacidade do sistema tratar de forma flexível às configurações da barras.

Dentre as funcionalidades descritas pelos autores não se destacam funcionalidades que facilitem a análise dos dados, assim como, a comparação com resultados de outros momentos, ou configurações, do sistema de potência.

Em [MS03] é proposto um sistema que permite ao usuário desenhar os modelos, inserir barras e configurar os elementos do sistema, a simulação e análise dos resultados são feitas através da interação com outros softwares, inclusive comerciais. Essa interação se dá por meio do banco de dados, escrevendo informações para serem acessadas de outros sistemas ou lendo informações escritas por outros sistemas, além da comunicação por meio de arquivos textos.

A principal função deste sistema é trabalhar com diagramas unifilares, porém faltam funcionalidades como a capacidade de exibição dos resultados das simulações junto ao diagrama,

²Do Inglês: *Integrated and Multi-purpose Package of Advanced Computational Tools for power system engineering*

assim como também, outras formas de análise dos resultados, a capacidade de realizar as próprias simulações.

Atualmente existem diversas ferramentas comerciais, dentre elas destacamos o *PowerWorld Simulator* da empresa *PowerWorld Simulator Corporation*³, que permitem a análise visual dos sistemas de potência. Essas ferramentas costumam apresentar um alto custo financeiro. No caso da ferramenta citada uma única licença do pacote básico custa \$8.000,00 (oito mil dólares) e pelo pacote completo paga-se \$84.000,00 (oitenta e quatro mil dólares) [Webe].

Outro software comercial é o *Power Systems Analysis Framework (PSAF)*⁴. Trata-se de um pacote integrado para análise de sistema de potência, constituído por uma interface gráfica com o usuário, sistema de banco de dados e um estimador de parâmetros dos elementos do sistema elétrico.

O pacote é constituído por diversos módulos de análise que são adquiridos em separado. Entre os módulos destacamos o módulo de Fluxo de Potência que possui uma rotina de cálculo baseado no método de Newton. Outro módulo é para análise de estabilidade de tensão que fornece gráficos para análise de curvas PV e QV. Apresenta também um módulo para análise da penetração de harmônicos nos sistemas de potência [webc].

O terceiro software comercial que destacamos brevemente é o *Interactive Simulation Package for Electrical Networks (ISPEN)*⁵ que consiste em uma família de softwares para simulação, planejamento e análise de contingências em sistema de potência, e apresenta uma rotina de cálculo que possibilita a análise de contingências em tempo real. O software fornece ferramentas análise de sensibilidade, análise de estabilidade, de risco, entre outras possibilidades [webb].

Outro software comercial trata-se do *PowerFactory Optimal Power Flow (OPF)*⁶ que busca solução de um fluxo de potência ótimo, procurando assim evitar a necessidade de interação entre o operados e os resultados obtidos, na busca de melhores valores de configurações para os parâmetros do sistema elétrico.

O sistema possui quatro opções de funções objetivo: minimização de perdas do sistema, minimização do custo de combustível para termoeletricas, maximização do lucro baseado nos custo

³Disponível em <http://www.powerworld.com/>

⁴Disponível em <http://www.powersys.fr/modulflow.php>

⁵Disponível em <http://www.eps.ch/50179596ad0c68204/index.html>

⁶Disponível em http://www.digsilent.de/Software/PowerFactory_Features/Optimal_Power_Flow/

de geração e de transmissão e a última opção de função objetivo consiste na minimização do intercâmbio entre as áreas[webd].

Outro software comercial trata-se do *EasyPower PowerFlow*⁷. Trata-se de um sistema de análise do fluxo de potência para redes de grande porte, onde as principais características destacadas pelo produtor do software são: uma interface gráfica com o usuário intuitiva, automatização de processos de análise e simulação interativa [weba].

2.4 Software Proposto

Neste trabalho foi desenvolvido um software para simulação de sistemas elétricos de potência, que permite a representação dos diversos parâmetros de controle que definem o estado de operação da rede, incluindo situações de contingências. Este software possibilita a simulação de sistemas de transmissão em diversas configurações de parâmetros para cargas, topologias e controles.

Os resultados das simulações são apresentados de maneira a oferecer ao usuário do sistema uma vasta gama de possibilidades de manipulação e interação com os dados. A implementação da interface com o usuário foi feita de forma a torná-la mais amigável, possibilitando assim, um uso intuitivo e claro do software, procurando facilitar as análises típicas e mais importantes, feitas por aqueles que trabalham com análise, planejamento e operação dos sistemas, com apoio nas soluções do fluxo de potência, tornando flexível a manipulação dos dados que compõem os sistemas [Lav00] [RB00] [HJAA98].

Para as simulações podem ser usados algoritmos presentes na literatura. Podemos citar o exemplo do algoritmo proposto por [Bau05], e codificado em linguagem C, para o cálculo de fluxo de carga em sistemas de transmissão, que é o algoritmo usado como padrão pelo software.

Também foi desenvolvido um conjunto de interfaces de software, que permite a criação de um padrão de comunicação com o sistema. A partir dessas interfaces é possível que algoritmos presentes na literatura sejam facilmente implementados e adaptados com o intuito de possibilitar a sua interação com o sistema.

⁷http://easypower.com/ep_powerflow.php

Obedecendo ao padrão definido, qualquer algoritmo pode ser inserido no sistema como um componente. Essa forma de desenvolvimento baseia-se no paradigma de softwares baseados em componentes.

Assim procuramos prever dentro do sistema desenvolvido uma flexibilidade que permite combinar algoritmos matemáticos, ferramentas para análise e visualização do sistema de potência tanto em forma texto, quanto através de recursos gráficos. Esses são os requisitos básicos necessários para um software moderno para área de potência [KML03].

2.5 Conclusão

Procurou-se neste capítulo descrever de forma genérica como uma visualização efetiva dos dados pode contribuir para a melhora no desempenho da análise e dos estudos dos mais diversos sistemas dando, enfoque para os sistemas elétricos de potência.

Foi feito também uma breve descrição de algumas das diversas proposta de softwares presentes na literatura que buscam trabalhar, sob os mais diversos aspectos, com os sistemas de potência. Foram descritos alguns software com enfoque comercial e quais os recursos que estes sistemas disponibilizam.

Apresentou-se uma breve descrição do software proposto nesta dissertação e no próximo capítulo será descrito os aspectos dos sistemas de potência que acreditam serem relevantes para o entendimento, de forma mais detalhada, de como o software proposto poderá auxiliar na análise e simulação dos sistemas de potência.

Capítulo 3

Formulação do Problema através de Cálculos Seqüenciais de Fluxo de Potência e a Estratégia do Software Interativo Proposto

Um sistema elétrico de transmissão conecta geradores e equipamentos de controle, através de linhas de transmissão e transformadores, aos sistemas de distribuição e demais cargas cujas demandas por energia devem ser atendidas. Em regime permanente os geradores são representados nas barras com tensão controlada (PV's) pela injeção líquida de potência ativa constante e pela magnitude da tensão sob controle constante. Os sistemas de distribuição e demais cargas que devem ser atendidas são representados por barras (PQ's) com injeções líquidas de potência ativa e reativa constantes. Os elementos de transmissão, constituídos por linhas de transmissão e transformadores, são representados pelos seus parâmetros juntos aos modelos de circuitos elétricos equivalentes [Saa02] [GSJ94] [Mon83].

Pode-se dizer que existem dois principais objetivos quando se trata de estudos de fluxo de potência em sistemas de transmissão. O primeiro diz respeito à necessidade de planejar as futuras expansões da rede e o segundo objetivo é descobrir a melhor maneira de operar o sistema existente [SJ82] [KML03]. As duas principais informações obtidas através destes estudos são o

módulo e a fase dos fasores da tensão nas barras do sistema e os fluxos de potência ativa (MW) e reativa (MVAR) nas linhas de transmissão e transformadores.

Neste capítulo é apresentado o problema de cálculo do fluxo de potência composto por um sistema de equações algébricas não lineares parametrizadas em variáveis controladas no sistema elétrico. Para atender restrições operacionais do sistema elétrico, as soluções dessas equações devem satisfazer um conjunto de desigualdades relacionadas aos valores de limites operativos dos equipamentos. No caso de infactibilidade dessas restrições a busca de alterações nas variáveis controladas do sistema é feita através de simulações baseadas na experiência dos planejadores e nas informações externas ao modelo matemático.

Na Seção 3.1 são apresentadas as equações do fluxo de potência, formalizando o problema em que o software proposto procura auxiliar na resolução e análise. Na Seção 3.2 são apresentadas algumas restrições operacionais dos sistemas de potência, mostrando assim como o software proposto procura servir de auxílio na solução destas restrições. Na Seção 3.3 é apresentada a estratégia de soluções para diversos problemas que podem surgir no planejamento e operação dos sistemas de potência e como o software proposto faz parte desta estratégia. Na Seção 3.4 são apresentadas as conclusões dos estudos feitos neste capítulo.

3.1 Equações do Fluxo de Potência

As barras, ou nós, que compõem o sistema podem ser divididas em dois grupos principais: as barras geradoras (PV's) e as de carga (PQ's). Além desses grupos há uma barra especial denominada de *slack* (SL) que tem como função principal servir de referência angular para o sistema. Nesta o ângulo de tensão nodal (θ) é prefixado. Assim as barras representam dois dos três principais componentes dos sistemas de transmissão de potência: os geradores e as cargas que estão agregadas ao sistema representando também os sistemas de distribuição. A topologia e os parâmetros dos circuitos de interligação das barras são representadas através da matriz admitância nodal [Saa02] [SJ82].

Tem-se no diagrama unifilar da Figura 3.1 a representação das barras de carga e geração, assim como a representação dos elementos, ramos, que são responsáveis pela interligação dessas barras.

Os geradores de energia, barras PV's, são responsáveis pela inserção de potência na rede e os nós PQ's são responsáveis pelas cargas consumidas mantendo o sistema em equilíbrio. Dessa forma as potências inseridas devem ser iguais às potências consumidas.

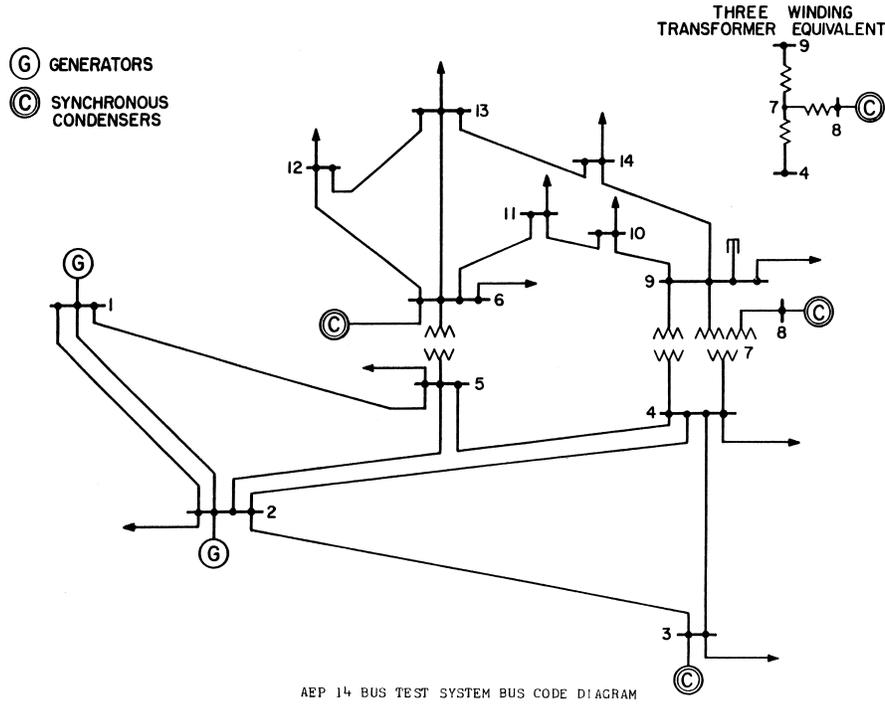


Figura 3.1: Figura do caso de teste IEEE de 14 barras, obtido de [IEE]

Os valores especificados para as injeções líquidas de potência ativa e reativa para cada barra, PV ou PQ, são representados pela Equação 3.1.

$$\begin{cases} P_k^{esp} = P_k^{Ger} - P_k^{Carga} \\ Q_k^{esp} = Q_k^{Ger} - Q_k^{Carga} \end{cases} \quad (3.1)$$

onde, os valores para P_k^{Ger} que representam a potência ativa gerada na barra k , P_k^{Carga} que representam a potência ativa consumida na barra k , Q_k^{Ger} que representam a potência reativa gerada na barra k e Q_k^{Carga} que representam a potência reativa consumida na barra k , são armazenados em um sistema de banco de dados como será descrito na Seção 4.4.1.

O problema do cálculo do fluxo de potência pode ser decomposto em dois subsiste-

mas [Mon83], o primeiro é representado pelas equações em 3.2 e o segundo é representado pela Equação em 3.3.

Pela resolução do subsistema (I) em 3.2 procura-se obter os ângulos das tensões nodais (θ_k) para as barras PV's e PQ's e os valores das magnitudes de tensões nodais (V_k) para as barras PQ. A partir das tensões ((V_k, θ_k)) atribuídas às barras do sistema, representadas nos respectivos vetores V e θ , e das expressões referentes às injeções líquidas de potência ativa ($P_k(V, \theta)$) e reativa ($Q_k(V, \theta)$) para as barras PQ e ativa ($P_k(V, \theta)$) para as barras PV é estabelecido o subsistema (I) [Mon83] das equações do fluxo de potência como descrito em 3.2. As magnitudes de tensão para as barras PV's também são especificadas e mantidas constantes.

$$(I) \begin{cases} P^{esp} - P_k(V, \theta) = 0, k \in \{PV\} \cup \{PQ\} \\ Q^{esp} - Q_k(V, \theta) = 0, k \in \{PQ\} \end{cases} \quad (3.2)$$

onde:

$$\begin{cases} V_k \in V, k \in \{PQ\} \text{ e } \theta_k \in \theta, k \in \{PQ, PV\} \text{ devem ser calculados.} \\ V_k \in V, k \in \{PV, SL\} \text{ são especificados.} \end{cases}$$

Pela resolução do subsistema (I), representado pelas equações em 3.2, obtém-se as magnitudes de tensão (V_k) e os ângulos (θ_k) nodais para todos os nós. As demais variáveis do sistema são obtidas através da resolução do subsistema (II) descrito em 3.3.

$$(II) \begin{cases} P_k(V, \theta) = V_k \sum_{l \in K} V_l [G_{kl} \cos(\theta_{kl}) + B_{kl} \sin(\theta_{kl})] \\ Q_k(V, \theta) = V_k \sum_{l \in K} V_l [G_{kl} \sin(\theta_{kl}) - B_{kl} \cos(\theta_{kl})] \end{cases} \quad (3.3)$$

onde G_{kl} e B_{kl} são a parte real e imaginária do elementos Y_{kl} da matriz admitância nodal, $[Y_{bus}]$ do sistema. Esses elementos são representados por $Y_{kl} = G_{kl} + jB_{kl}$, $j = \sqrt{-1}$. Na matriz $[Y_{bus}] = [G] + j[B]$ estão representados a topologia e os parâmetros do sistema elétrico de transmissão de potência.

A partir da solução do sistema de equações descrito por 3.2 e 3.3 é possível determinar os fluxos de potência ativa e reativa através dos circuitos. São no total quatro variáveis para cada circuito, como pode ser observado na Figura 3.2, a potência ativa no sentido da barra k para a barra l e no sentido reverso, descritas por P_{kl} e P_{lk} , respectivamente, as outras duas variáveis dizem respeito à potência reativa, descritas por Q_{kl} e Q_{lk} .

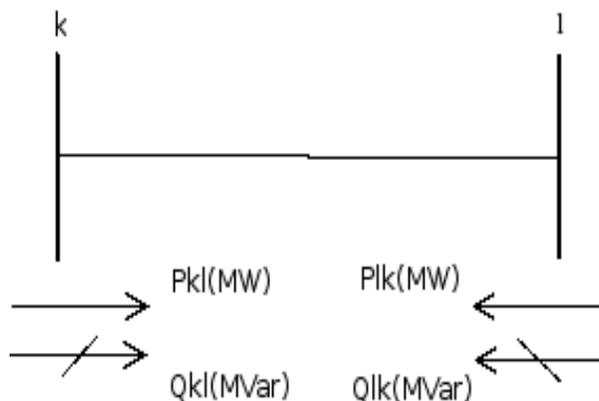


Figura 3.2: Fluxo de potência ativa e reativa em um ramo

Os cálculos dos fluxos de potência nos circuitos são feitos através das equações simples que se utilizam de parâmetros e das tensões de seus nós terminais[Mon83].

3.2 Restrições operacionais e limites a serem atendidos

Um sistema de transmissão pode operar sob várias configurações que se caracterizam por especificações nas variáveis de controle, níveis de carga pré-fixados os em determinados valores e na topologia da rede. Os níveis de carga se apresentam nos parâmetros das barras PQ, por exemplo, e são definidos de acordo com as necessidades do mercado, ou seja, dos consumidores de energia. Os parâmetros das barras PV's são dados pela geração de potência ativa (MW) definidas pelo pré-despacho e no caso das magnitudes de tensão, pelos operadores. Simulações de variações nesses parâmetros, com o software proposto, podem ser realizadas com os dados de cada uma das 24 horas do dia, desconsiderando a transitoriedade do sistema.

Outro fator a ser considerado, e que pode gerar novas configurações, é a mudança na topologia da rede. Estas mudanças de topologia dizem respeito às contingências que um sistema

está sujeito, ou seja, a perda de uma linha ou mudanças severas em suas características, o que acaba por influenciar os fluxos de potência em todo o sistema. Nestes casos a matriz admitância nodal, $[Y_{bus}]$, do sistema é modificada para passar a representá-lo sem o elemento contingenciado.

As mudanças nos dados do sistema podem ocorrer não só pela necessidade dos consumidores de energia, mas também para se obedecer a restrições operacionais. Para isso, inequações adicionais às equações do fluxo de potência devem ser consideradas. Estas restrições podem ser tanto de operação quanto de atuação dos dispositivos de controle. Os limites podem ser, por exemplo, limite de injeção de potência reativa (Q_k) para as barras PV's, limites para as magnitudes de tensão nodal (V_k) nas barras PQ e PV e limites nos fluxos de potência ativa em um circuito. As inequações que representam estas restrições são respectivamente 3.4, 3.5 e 3.6.

$$V_k^{min} \leq V_k \leq V_k^{max}, k \in \{PQ\} \quad (3.4)$$

$$Q_k^{min} \leq Q_k(V, \theta) \leq Q_k^{max}, k \in \{PV\} \cup \{SL\} \quad (3.5)$$

$$\|P_{kl}(V, \theta)\| \leq P_{kl}^{max} \quad (3.6)$$

onde, P_{kl} é o fluxo de potência ativa saindo do nó k em um circuito do ramo kl .

Para poder validar o sistema elétrico nas mais diversas configurações, simulações devem ser feitas. É possível, com o sistema proposto, criar cenários para cada configuração, simular, analisar os resultados e compará-los com os resultados de outras configurações, armazenando os dados relativos aos cenários operativos viáveis no banco de dados.

Quando temos uma barra de carga (PQ) no qual o valor da magnitude de tensão nodal (V_k) não respeita os limites definidos, representados pela Inequação 3.4, uma possível medida a ser adotada pelo operador do sistema é, por exemplo, procurar na vizinhança da respectiva barra PQ uma barra com controle de tensão (PV) e, nesta, atuar para corrigir aquela. Isso pode ser feito, por exemplo, através da funcionalidade descrita na Seção 5.1.1.

Ao encontrar uma ou mais barras PV's na vizinhança mais próxima da barra de carga

(PQ), com tensão não factível, o operador pode alterar as magnitudes de tensão das barras PV's. Caso a tensão da barra de carga (PQ) esteja menor do que o limite inferior, será testado uma solução com aumento da tensões das barras PV's vizinhas. Ao realizar estas alterações é possível simular a nova configuração e verificar o perfil de tensão do sistema.

Com as soluções armazenadas no banco de dados é possível comparar esta nova configuração com as configurações anteriores. Observando-se assim não só as alterações que ocorreram na barra com problemas, mas também os efeitos das alterações em todos os outros elementos do sistema. Estas comparações podem ser feitas através dos recursos gráficos descritos na Seção 5.1.2.

A manutenção do perfil de tensão nas barras PQ's dentro dos limites é importante, por exemplo, no momento do planejamento da expansão do sistema, para verificar como a inserção de novos nós e circuitos devem ser realizadas para que a tensão nodal (V_k) continue respeitando seus limites operacionais [Mon83].

O software proposto permite que sejam inseridos novos elementos no sistema de acordo com as funcionalidades descritas na Seção 5.1.1. Nestes casos, ao se realizar uma simulação a matriz de admitância nodal, $[Y_{bus}]$, do sistema é reformada para passar a representar os novos elementos do sistema.

Exibindo o perfil de injeção de potência reativa para as barras PV's por meio da funcionalidade descrita na Seção 5.1.2, por exemplo, é possível observar quais barras estão infactíveis, ou próximas da infactibilidade, de acordo com a Inequação representada em 3.5. A correção de uma situação como esta passa por alterar a tensão especificada da própria barra infactível. Assim, após essa alteração passa-se a realizar uma nova simulação, não só para verificar a respectiva injeção de potência reativa, mas também para verificar como a alteração na tensão da barra PV alterou o perfil de tensão das outras barras do sistema.

Como definido na Inequação 3.6, existem restrições operacionais para a capacidade de potência que um circuito é capaz de transportar. Nestes casos é necessário alterar a geração de potência nos geradores e realizar novas simulações para verificar se nenhum outro elemento do sistema possui infactibilidades.

Outras simulações podem ser realizadas alterando-se os valores dos dados dos circuitos, por exemplo, pode ser feito o estudo de como a variação da admitância de um ramo afeta a distribuição de correntes pelo sistema elétrico. É possível também utilizar a variação do *tap* dos

transformadores para verificar as alterações nos perfis de tensão do sistema.

3.3 Estratégia do Software Interativo

A finalidade do software interativo desenvolvido é permitir que especialistas tenham a oportunidade de, por meio de procedimentos ágeis para a realização de simulações, obter cenários operativos viáveis que atendam ao modelo do fluxo de potência representado pelas equações em 3.2 e aos limites operacionais representados pelas Inequações 3.4, 3.5 e 3.6.

Um cenário operativo é caracterizado por:

1. um conjunto de variáveis controladas que será representado por um vetor definido por:

$$u = \begin{cases} P_k^{Ger}, k \in \{\text{Barras de Geração}\} \\ V_k, k \in \{SL, PV\} \end{cases} \quad (3.7)$$

2. uma configuração topológica com os respectivos parâmetros dos circuitos do sistema considerado nas equações de potência (3.3) por meio dos elementos das matrizes $[G]$ e $[B]$ do sistema e aqui representadas simbolicamente como p ;
3. uma configuração de cargas representadas como potências definidas em 3.1 como P_k^{Carga} e Q_k^{Carga} aqui também considerados simbolicamente em p , como no item b);
4. perfil de tensão (V_k) das barras de carga (PQ) calculado a partir das equações (I) do problema do fluxo de potência (3.2) e representado por um vetor como:

$$x = [V_k, k \in \{PQ\}] \quad (3.8)$$

5. desse modo um cenário operacional se caracteriza pela definição de u^{def} e p^{def} por parte dos especialistas planejadores e pelo cálculo de x a partir das equações (I) do problema do fluxo de potência (3.2) que passam a ser representada como:

$$g(x, u^{def}, p^{def}) = 0 \quad (3.9)$$

cuja solução x^{cal} é calculada por um algoritmo baseado no método de Newton[TH57] [Sto74] [Mon83].

O cenário operacional definido por $(x^{calc}, u^{def}, p^{def})$ pode atender ou não os limites representadas pelas Inequações 3.4, 3.5 e 3.6 que passam a ser representadas por 3.10. E esta inequação será resolvida pelas interações entre o especialista, que altera u^{def} e p^{def} , e o software interativo que calcula as novas soluções, tendo como facilidade os recursos gráficos disponíveis no sistema proposto.

$$\left\{ \begin{array}{l} h(x^{calc}, u^{def}, p^{def}) \leq 0 \\ OU \\ h(x^{calc}, u^{def}, p^{def}) \approx 0 \end{array} \right. \quad (3.10)$$

A estratégia para a geração de cenários operativos factíveis através do software interativo proposto pode ser ilustrado pelo diagrama apresentado na Figura 3.3.

Abaixo segue a descrição detalhada de cada um dos passos descritos na Figura 3.3:

1. As infactibilidades presentes em 3.10 são traduzidas em termos gráficos por meio das interfaces interativas, permitindo que ações sejam tomadas sobre as variáveis de controle, as configurações de carga e, caso necessário, sejam feitas alterações na topologia da rede, que também é exibida graficamente pelo sistema;
2. É especificada uma proposta de configuração inicial (u^{def} e p^{def}) fornecida pelos especialistas e pelas necessidades do mercado;
3. A solução do fluxo de potência se dá de acordo com a Equação em 3.9;
4. É feita a análise das soluções operativa, através das disponibilidades das interfaces interativas, verificando-se a observância, ou não, dos critérios definidos em 3.10, identificando possíveis elementos infactíveis;
5. A partir da análise das variáveis e das possibilidades de alterações nos diversos elementos do sistema, com as ferramentas fornecidas pela interface interativa, o operador do sistema

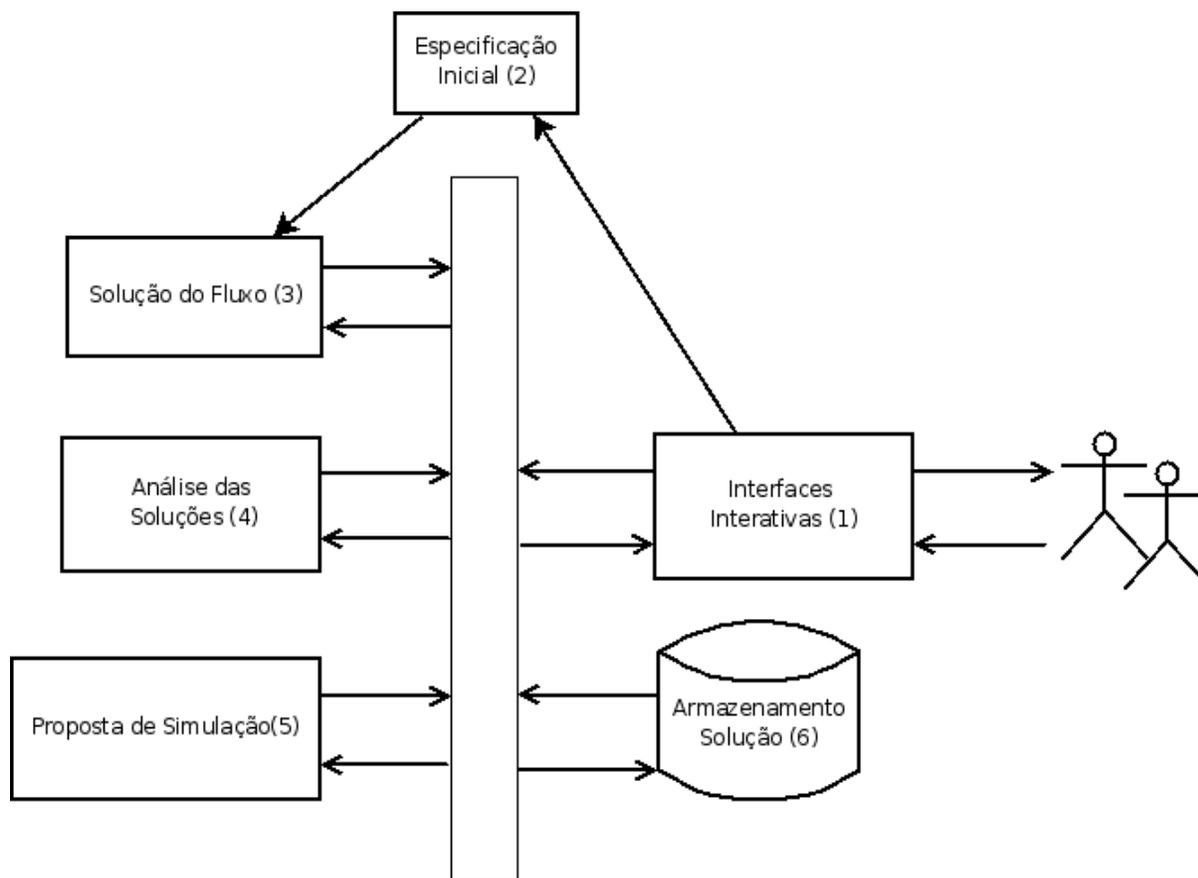


Figura 3.3: Descrição da Estratégia do Interação

propõe uma nova configuração baseada na sua experiência. Esta nova configuração será simulada e terá seus resultados analisados e assim sucessivamente;

6. Caso o resultado da simulação seja factível e/ou seja de interesse do operador, é possível armazenar a configuração e os dados da simulação no banco de dados.

De uma maneira geral, todas as variáveis dos elementos que compõem o sistema elétrico podem ter seus valores alterados. Ao se considerar um novo sistema elétrico o software cria no banco de dados um caso base, ou uma configuração base. Assim, quando qualquer alteração nos dados ou na topologia da rede é feita o software pode armazenar estas alterações no banco de dados, como uma subconfiguração do caso base. Toda simulação realizada é associada a uma configuração ou subconfiguração, permitindo relacionar os dados obtidos nas simulações com os dados fornecidos como parâmetros.

No software proposto há possibilidade da inserção de outros algoritmos para o cálculo do fluxo de potência, permitindo a comparação não só de configurações diferentes da rede elétrica como também de implementações e abordagens diversas para a obtenção dos resultados.

No protótipo implementado já existe um algoritmo inserido por *default*. Este algoritmo, descrito em [Bau05], obtém a solução do sistema especificado em 3.2 pelo método de Newton [TH57] [Sto74] [Mon83] , e as tensões obtidas para as barras de carga devem ser mantidas em níveis previamente especificados [Sar02].

3.4 Conclusão

Neste capítulo procura-se estabelecer uma relação entre os elementos que compõe um sistema elétrico de potência e as possibilidades de alterações dos parâmetros desses elementos na busca de uma configuração eficiente para o funcionamento do sistema.

Procura-se explicar como se dá o processo de interação entre o software proposto e um especialista que, usando sua experiência, busca as configurações mais adequadas de funcionamento.

As descrições deste capítulo foram feitas tendo com foco o modelo de representação do sistema de potência. No próximo capítulo descrevemos quais elementos e estruturas de software foram propostos para que os objetivos aqui descritos fossem alcançados, enfocando neste caso, os aspectos computacionais.

Capítulo 4

Projeto do sistema

Neste capítulo serão destacados os aspectos computacionais do software proposto. O sistema foi todo projetado usando orientação a objetos, uma vez que para o desenvolvimento de softwares específicos para sistema de potência, a orientação a objetos trás diversas vantagens como: a maior facilidade em se escrever código reusável, o uso de objetos permite modelar sistemas complexos, como são os sistemas de potência [KML03].

Foram utilizadas bibliotecas e ferramentas computacionais que não gerassem custos financeiros na aquisição de licenças. Com a descrição dos recursos computacionais utilizados, junto com um detalhamento da estrutura do software, procura-se facilitar processos de evolução do sistema proposto.

Este capítulo esta dividido em quatro seções. Na Seção 4.1 serão apresentados conceitos básicos de Orientação a Objetos. Na Seção 4.2 serão apresentados os recursos de software utilizados para o desenvolvimento do protótipo do sistema proposto. Na Seção 4.3 serão discutidas as funcionalidades do sistema de acordo com a visão do usuário. Diversas classes foram definidas, de forma a dar um tratamento adequado ao problema de criação de um sistema de visualização efetiva dos dados resultantes das simulações em sistemas de potência, de acordo com a visão do usuário discutida em 4.3. Os conjuntos de classes implementadas serão brevemente discutidas na Seção 4.4.

4.1 Conceitos de Orientação a Objetos

É necessário primeiramente à definição de alguns conceitos utilizados em projetos de sistemas orientados a objetos. O primeiro conceito refere-se a classe, que é responsável por encapsular os dados e os procedimentos necessários para descrever as propriedades e o comportamento de uma entidade [Pre95].

Os dados presentes em uma classe são ditos encapsulados por não poderem ser manipulados diretamente, mas somente através dos procedimentos, operações ou serviços fornecidos na interface da classe. De outra maneira uma classe pode ser vista como uma descrição generalizada de um conjunto de objetos [Pre95].

Uma superclasse, ou *classe pai*, é uma classe da qual outras são derivadas, estas podem ser chamadas de *classes derivadas* ou *classes filhas*. Dessa maneira pode-se construir, por meio da classe pai, uma interface genérica que é então especializada através das classes filhas.

Uma classe pode ser classificada, ainda de diversas maneiras, como concreta e abstrata, por exemplo. O conceito de classe abstrata diz respeito a impossibilidade de se instanciar um objeto a partir desse tipo de classe, enquanto uma classe concreta permite essa criação de forma direta. Dessa uma classe abstrata só pode funcionar como uma superclasse, onde suas classes derivadas, caso não sejam abstratas, podem ser instanciadas.

Uma classe abstrata é aquele que possui métodos abstratos, ou seja métodos que não foram implementados, cabendo esta implementação as classes derivadas e estas, só se tornam concretas, caso implementem todos os métodos abstratos de suas classes pais que ainda não o foram. Assim uma classe abstrata é aquela criada com o intuito de possuir especializações [RJB04].

Outro conceito importante diz respeito à herança, onde uma subclasse pode herdar total ou parcialmente os métodos e atributos associados com a superclasse, isso faz com que o reuso de código se de maneira direta e transparente.

A consistência de objeto em termos computacionais, se assemelha ao que é um objeto no mundo real, ou seja, ambos são compostos por estados e comportamentos, que consistem, respectivamente, em termos computacionais as propriedades e ao métodos dos objetos. Um objeto é a instância de uma classe [RJB04].

Além dos conceitos de Orientação a Objetos, é necessário definir alguns conceitos básicos de UML¹, que trata-se de uma linguagem de modelagem que procura ajudar a descrever e projetar softwares, em particular aqueles construídos de acordo com os preceitos da orientação a objetos [Fow03].

O primeiro conceito diz respeito aos casos de uso, que em UML, são interações típicas entre o usuário e o sistema. Os conceitos principais em casos de uso são: cenário que consiste no conjunto de passos da interação entre o sistema e o usuário e, o outro conceito, é o ator, que é a representação dos diversos usuários do sistema [Fow03].

Nas subseções da Seção 4.4 serão mostrados diversos diagramas, em UML, de classes do sistema. Que consiste em um dos tipos de diagrama mais amplamente utilizados e que descreve os tipos de objetos do sistema e como estes objetos se relacionam estaticamente. Fornecendo, assim, uma maneira de se ter um bom entendimento do funcionamento do sistema [Fow03].

Será apresentado, também, na Seção 4.4 o diagrama, em UML, de pacotes do sistema. Este diagrama, normalmente, é usado para agrupar classes, facilitando a estruturação de grandes sistemas [Fow03].

4.2 Recursos Computacionais Utilizados

O protótipo proposto foi implementado em C++, por ser uma linguagem que em essência procura facilitar o processo de construção de grandes e complexos sistemas [Sch98]. A linguagem também permite a implementação dos principais princípios da orientação a objeto, que são o encapsulamento, polimorfismo e herança, suportando abstração de dados e a programação genérica [Str97].

Foram utilizados recursos e bibliotecas distribuídas de acordo com a licença GPL², ou distribuídos de acordo com uma de suas variantes. A implementação do sistema utilizou-se de diversas ferramentas como o *gprof*³ que possibilita analisar o desempenho de uma aplicação, identificando as porções de código que mais consomem tempo de execução. Com esta informação

¹Do Inglês: Unified Modeling Language

²Do Inglês: *Gnu Public License* disponível em <http://www.gnu.org/licenses/gpl.html>

³Referência em http://www.gnu.org/manual/gprof-2.9.1/html_mono/gprof.html

é possível otimizar somente as partes do código que terão alguma relevância na performance geral da aplicação [Jon05].

Outra ferramenta utilizada foi o compilador `g++`⁴. Que, na verdade, é o compilador para linguagem C++ do pacote *GCC* que consiste em um poderoso conjunto de recursos capaz de construir um binário a partir do código fonte de alto nível, oferecendo suporte para diversas linguagens, como: C, C++, Ada, Java, Fortran, entre outras. Este compilador suporta mais de 40 diferentes famílias arquiteturas, isso facilita o processo de *cross-compilação*, que consiste em uma arquitetura compilar um binário que é válido para outra arquitetura. [Jon05] [LO97].

Para auxiliar no processo de busca de falhas de implementação do protótipo proposto fizemos uso da ferramenta conhecida como *GDB*⁵, um poderoso *depurador* com muitas funcionalidades, como é característico da grande maioria das ferramentas de código aberto, dentre elas podemos citar a capacidade de trabalhar com processos já iniciados, alterar o valor das variáveis durante o processo de execução e navegar pela pilha de chamadas do processo [LO97].

As três ferramentas citadas anteriormente fazem parte do *GNU*⁶ que é um projeto com mais de 20 anos que tem como principal objetivo desenvolver um sistema operacional *Unix-like* com a característica de que o conjunto de ferramentas desenvolvidas sejam desenvolvidas e mantidas com o código aberto.

Outro recurso utilizado na implementação foi o software *Valgrind*⁷, que fornece muitos recursos para o processo de correção de erros, assim como para o processo de detecção de falhas no gerenciamento da memória e falhas na construção ou controle de *threads*. A principal *IDE*⁸ utilizada no desenvolvimento foi o *Eclipse*⁹ com *plugin CDT*¹⁰.

Todo o sistema utiliza os recursos disponíveis na *STL*¹¹ que esta no padrão *ANSI/ISO* para a linguagem C++. A decisão pelo uso dos recursos dessa biblioteca se deu pelo fato dela tornar a programação mais produtiva, uma vez que ela é formada por um conjunto de *containers* e

⁴Disponível em <http://gcc.gnu.org/>

⁵Disponível em: <http://www.gnu.org/software/gdb/>

⁶Disponível em: <http://www.gnu.org/>

⁷Disponível gratuitamente em: <http://valgrind.org>

⁸Do inglês: *Integrated Development Environment*

⁹Disponível gratuitamente em: <http://www.eclipse.org/>

¹⁰Disponível em : <http://www.eclipse.org/cdt/>

¹¹Do Inglês: *Standard Template Library*

de algoritmos genéricos, que trabalham com boa parte dos *containers* disponíveis, facilitando uma programação mais genérica [MDS01].

As principais bibliotecas utilizadas foram o *Gtk+*¹², através de sua interface oficial para C++ *Gtkmm*¹³ e a biblioteca *Boost*¹⁴. A biblioteca *Gtk+* foi escolhida por se tratar de um poderoso e muito usado *toolkit* para criar interfaces gráficas com o usuário. O *Gtk+* é estruturado para ser multi-plataforma e possui uma vasta e completa documentação disponível na Internet [Pen99].

A biblioteca *Boost* é na verdade um conjunto de bibliotecas submetido ao comitê de padronização da linguagem C++, ela também é construída buscando ser multi-plataforma. A principal vantagem em se usar os recursos desta biblioteca é, que devido às características intrínsecas do modelo de desenvolvimento do software aberto, as soluções fornecidas pela biblioteca tendem a serem as melhores possíveis, reduzindo a probabilidade de erros, os custos de manutenção e aumentando a produtividade do desenvolvimento, uma vez que, tendo bibliotecas já implementadas e testadas, basta usá-las.

Dentre as diversas funcionalidades disponíveis na biblioteca *Boost*, pode-se citar algumas como: *CRC*, *FileSystem* que fornece um conjunto de facilidades para manipular o sistema de arquivo, *GIL* que consiste em uma biblioteca genérica para imagens, *Graph* que procura ser uma espécie de *STL* para grafos, dentre dezenas de outras bibliotecas. No protótipo implementado foram utilizadas as bibliotecas que fornecem funcionalidades para manipulação de data e hora, criação de *threads* e manipulação de sinais.

Como as bibliotecas citadas anteriormente são disponibilizadas em código aberto é facilitado o processo de se portar para outras plataformas, assim, apesar do protótipo ter sido construído em um sistema *GNU/LINUX* em linguagem C++ a compilação para o sistema operacional *Windows Xp*, por exemplo, se dá de maneira transparente em termo do código, ou seja, o código que gera o binário executável no sistema *GNU/LINUX* não precisa ser alterado, por requisitos de plataforma, para gerar o binário executável no sistema operacional *Windows Xp*, no qual é possível realizar o desenvolvimento através da IDE *Visual C++*¹⁵ [Mue02].

¹²Disponível em: <http://www.gtk.org>

¹³Disponível em: <http://www.gtkmm.org>

¹⁴Disponível em: <http://www.boost.org>

¹⁵Disponível em: <http://www.microsoft.com/express/vc/>

4.3 Visão do Usuário

Conforme pode ser observado na Figura 4.1, o sistema fornece seis grupos de funcionalidades principais. São elas: inserção de modelos de sistemas elétricos, controle de simulações, controle dos momentos ou estados do sistema elétrico, inserção de algoritmos para a realização das simulações, apresentação dos resultados e interação com os modelos de sistemas elétricos disponíveis.

A partir deste momento no texto passa-se a nos referir aos algoritmos implementados para a realização das simulações como módulos e no referiremos aos estados que um sistema elétrico pode assumir como configurações ou momentos do sistema.

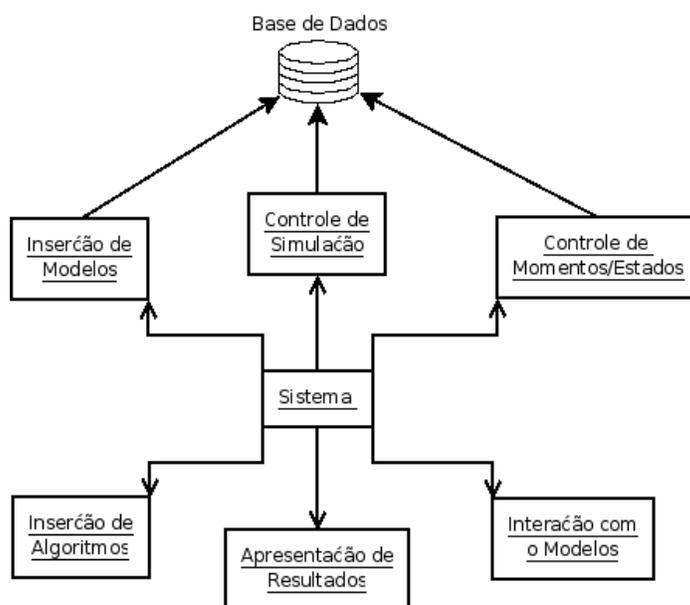


Figura 4.1: Visão do usuários das funcionalidades do sistema

O sistema conta com o apoio de um bando de dados relacional. Atualmente o sistema gerenciador de banco de dados (SGBD) utilizado é o *Firebird*¹⁶, porém foi projetada uma camada de software que permite independência quanto ao SGBD utilizado.

A forma como o modelo do banco de dados utilizado foi construído, permite que o armazenamento das configurações do sistema, assim como das simulações realizadas para cada

¹⁶Este SGBD pode ser obtido gratuitamente em <http://www.firebirdsql.org/>

configuração. Existe no modelo uma tabela para cada tipo de elemento do sistema elétrico tratado, assim como tabelas que permitem armazenar e correlacionar os dados das configurações e das simulações com os respectivos elementos. É feito também o armazenamento dos modelos dos sistemas elétricos inseridos, assim como, o armazenamento das informações necessárias para carregar os módulos de cálculo. Outros aspectos do banco de dados são discutidos na Seção 4.4.1.

A funcionalidade que permite a inserção de novos modelos ao sistema, foi projetada para que essa inserção se dê através de arquivos em formato texto que descrevem o modelo. Foram inicialmente implementados suporte para o formato IEEE [IEE] e PTI [IEE], mais detalhes serão discutidos na Seção 4.4.2.

O controle de simulação permite que sejam acessadas as simulações já realizadas para cada modelo, assim como o controle das simulações que ainda estão ocorrendo, permitindo ao usuário a inserção de limite de tempo ou de iterações para a execução, mais detalhes desta funcionalidade serão discutidos nos tópicos 4.4.2 e 5.1.2.

Outro grupo de funcionalidades diz respeito ao controle das configurações do sistema, permitindo ao usuário a modificação, remoção e inserção de configurações para um determinado modelo de rede elétrica, estas funcionalidades também serão abordadas nos tópicos 4.4.2 e 5.1.1, sendo que neste último, serão abordados os resultados obtidos também para a funcionalidade de interações com o modelo, onde é possível a inserção de novos elementos, de contingências e as modificações dos dados de um modelo de rede elétrica que está sendo estudado pelo usuário do sistema.

A apresentação dos resultados consiste no grupo de interfaces de uma forma geral, mas com enfoque maior nos gráficos dos resultados, que permitem a análise de uma ou mais simulações, facilitando comparações. Esta funcionalidade será abordada nos tópicos 4.4.3 e 5.1.2.

A última funcionalidade da visão do usuário refere-se a inserção de novos módulos de cálculo, o que permite testes e comparações de algoritmos e rotinas de cálculo usando as ferramentas descritas anteriormente. Esta funcionalidade será descrita nos tópicos 4.4.6 e 5.3.

4.4 Diagramas de Projeto

De acordo com as funcionalidades descritas na Seção 4.3 e de acordo com a necessidade de classes para implementação destas funcionalidades foi definido um conjunto de seis pacotes que agrupam as classes que possuem contratos de funcionalidades semelhantes e/ou complementares. O diagrama de pacotes do sistema proposto pode ser observado na Figura 4.2.

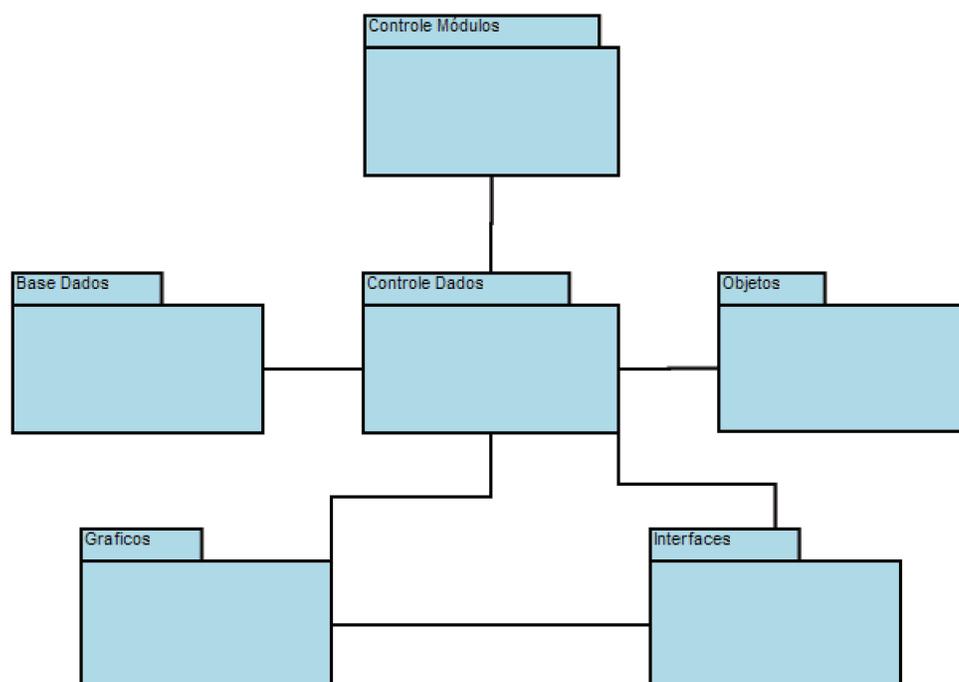


Figura 4.2: Diagrama de pacotes

Tem-se o pacote Controle de Módulos, que será detalhado na Seção 4.4.6, que contém as classes responsáveis pela inserção, identificação, remoção dos módulos no sistema. O pacote Base de Dados, discutido na Seção 4.4.1, e que apresenta as classes que implementam a camada de abstração entre a aplicação e o SGBD utilizado, permitindo que esta escolha fique transparente para o sistema.

O pacote de Controle de Dados, discutido em 4.4.2, contém as classes que definem os objetos responsáveis pelo controle de dados, aí se inclui o controle de modelos de sistemas elétricos, suas configurações e simulações. No pacote Objetos, detalhado na Seção 4.4.4, tem-se as classes que representam os elementos dos sistema elétrico, além das estruturas para a manipulação

e organização deste elementos dentro da dinâmica do sistema.

Temos o pacote Graficos, discutido em 4.4.3, compostas pelas classes responsáveis pela preparação dos dados para que serão utilizados nos gráficos assim como a construção da interface gráfica com usuário destes gráficos. O último pacote refere-se as interfaces gráficas do usuário de uma forma geral, com exceção das presentes na Seção 4.4.3, e será discutida no Seção 4.4.5.

4.4.1 Base de Dados

Como pode ser observado no diagrama da Figura 4.3 o conjunto básico responsável pelo fornecimento de acesso ao banco de dados para todas as outras partes do sistema é formado por três classes. *PrepareSQL* é responsável por preparar as partes básicas de todo tipo de consulta SQL que pode ser feita à base de dados.

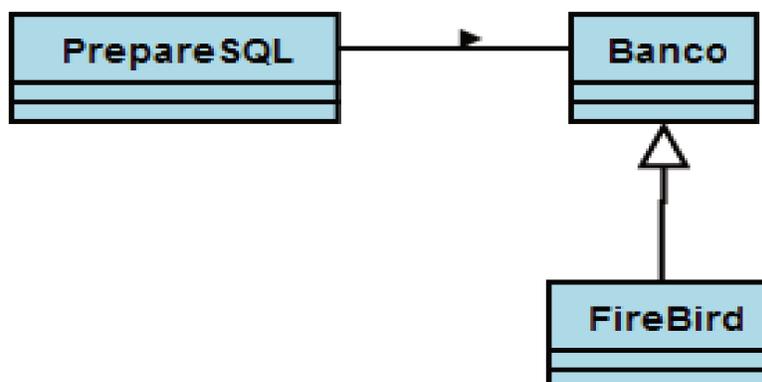


Figura 4.3: Diagrama de classes, visão banco de dados

A classe *Banco* fornece a interface para que todas as partes do sistema acessem o banco de dados independente do SGDB utilizado. Ela fornece os métodos básicos de conexão, criação e finalização de transações, execução de *statements* SQL. Esses métodos precisam ser implementados pelas classes que dela derivam, no diagrama temos a classe *fireBird*, único suporte implementado no protótipo, contudo, além das funcionalidades de abstração citadas até o momento, *Banco* fornece um conjunto de funções para a manipulação dos resultados obtidos através de qualquer consulta SQL.

O modelo de dados usado na construção do banco esta representado na Figura 4.4, usou-se uma modelagem relacional. Temos a tabela *sistemas* responsável por armazenar todos os dados

referentes a um sistema elétrico e para descrever os dados genéricos referentes às configurações, ou momentos, do sistema elétrico é utilizada a tabela *descrMomeCirc*.

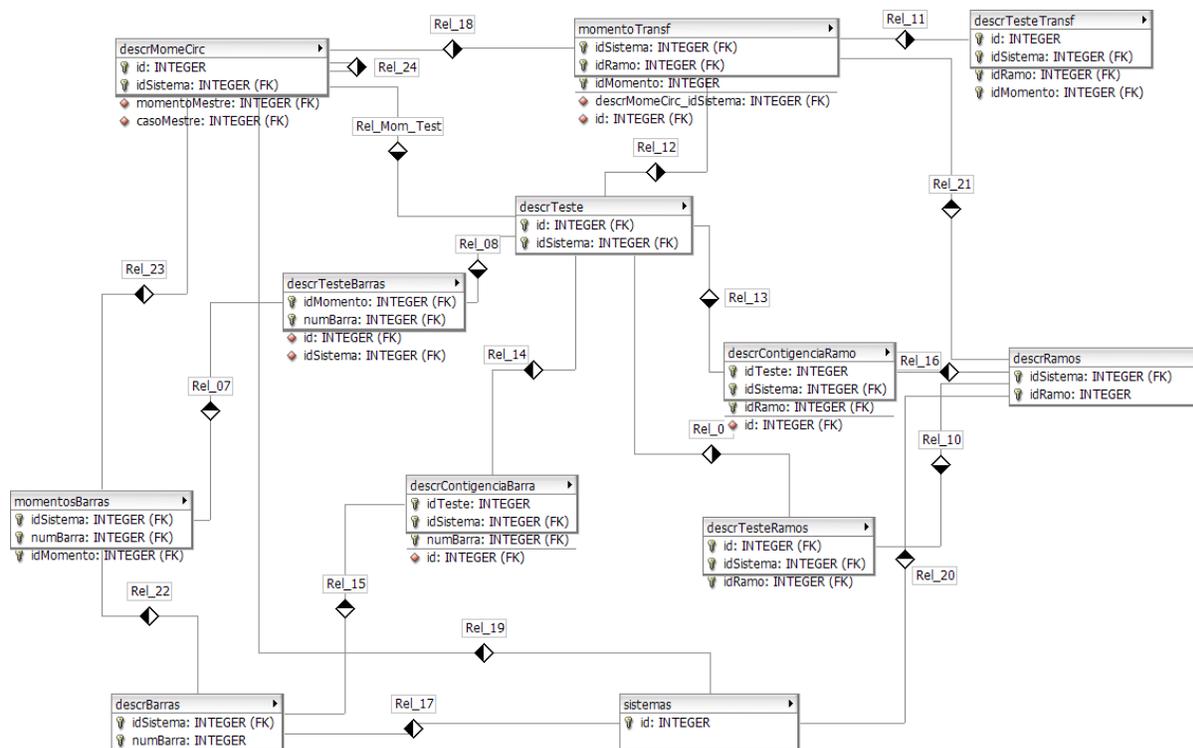


Figura 4.4: Modelo banco de dados

As informações referentes aos nós do sistema elétrico e que não se alteram nas configurações subsequentes são armazenadas na tabela *descrBarras*, o mesmo ocorre para o elementos de ligação, neste caso a tabela usada é a *descrRamos*.

As dados das configurações, ou momentos, do nó ficam armazenados em *momentosBarras* e para o elementos de ligação, e neste caso mais especificamente para os transformadores, a tabela utilizadas é a *momentoTransf*. As alterações na topologia da rede ficam armazenada em *descrContingenciaBarra*, para o nós, e *descrContingenciaRamo* para todos os elementos de ligação do sistema.

As informações gerais de uma simulação ficam armazenadas na tabela *descrTeste*, os dados obtidos pelas simulações são armazenadas de acordo com o tipo de elemento, assim temos

três tabelas para armazenar os dados numéricos: *descrTesteBarras* para armazenar os resultados das barras, *descrTesteRamos* para armazenar os resultados referentes às linhas de transmissão e *descrTesteTransf* que armazena os resultados referentes aos transformadores.

4.4.2 Controle de Dados

O pacote *Controle de Dados* é responsável pelo controle de dados do sistema e tem suas principais classes observadas no diagrama presente na Figura 4.5. Em termos de funcionalidades temos três classes principais: *BaseArquivo*, *ProblemaPrincipal* e *ConjuntoExecucao*.

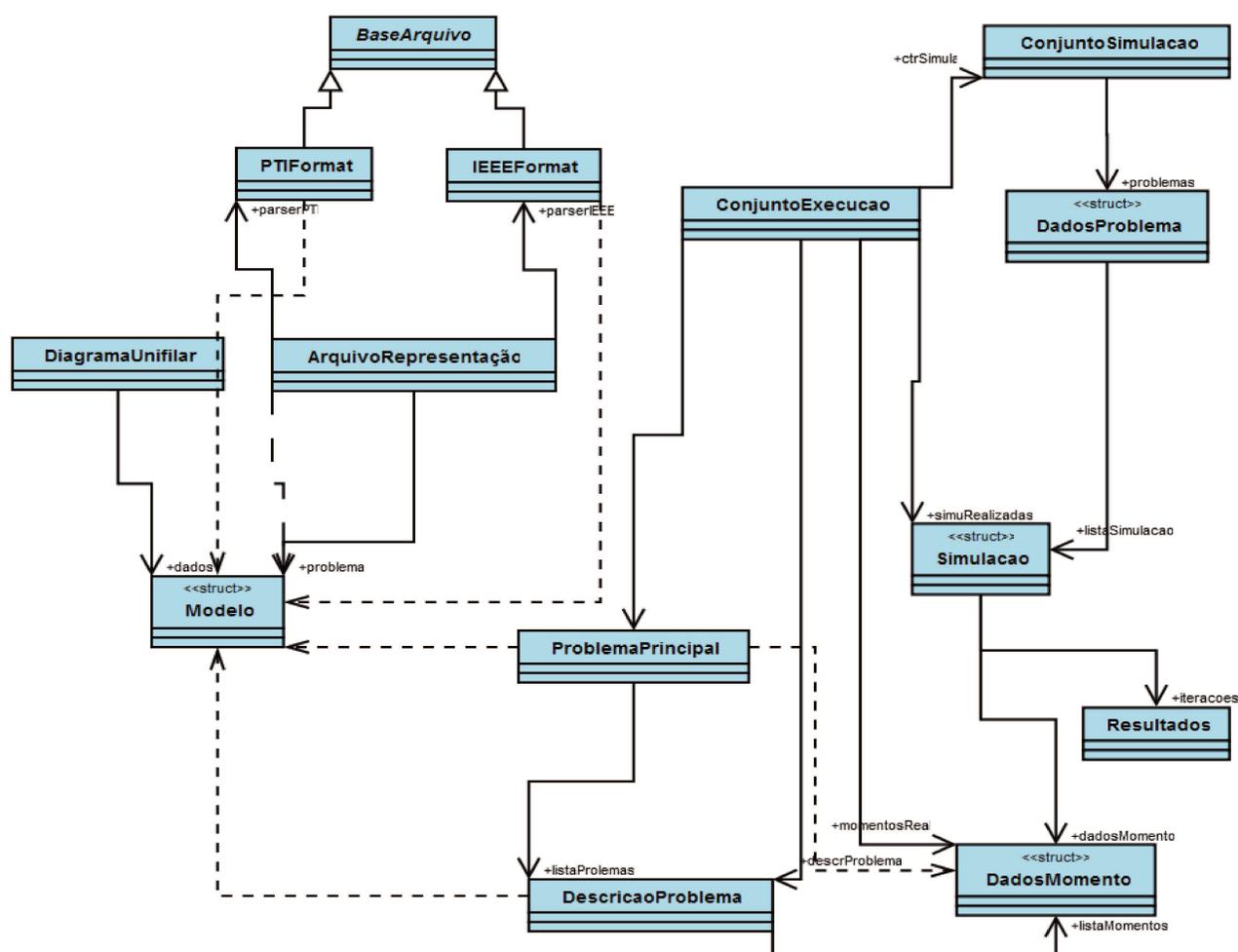


Figura 4.5: Diagrama de classes, visão controle de dados

A classe *BaseArquivo* fornece os métodos que permitem a manipulação dos arquivos em formato texto que contém a descrição de um modelo de um sistema elétrico. Para cada padrão de organização dos dados, nesta estrutura, deve ser implementada um classe que seja capaz de interpretar o arquivo, no modelo de classes (Figura 4.5) tem-se duas classes com esse propósito *PTIFormat* e *IEEEFormat*, que manipulam os dados no formato *PTI* e *IEEE* respectivamente.

Através da classe *ArquivoRepresentação* é fornecida mais uma camada de abstração capaz de identificar em qual padrão os dados do arquivo estão armazenados, fazendo a chamada para o decodificar adequado, mantendo isolado do resto do sistema todos os aspectos necessários para ser decodificar descritores de padrões de representação de sistemas elétricos, pode-se dizer que uma instância dessa classe fornece um objeto *façade* [GHRV00]. Esse classe obtém de *BaseArquivo* uma instância da estrutura *Modelo* e a repassa para quem a chamou, normalmente esta instância deve ser fornecida a classe *ProblemaPrincipal*.

A estrutura *Modelo* fornece os dados básicos de um modelo, como nome, descrição, número de barras, ramos e transformadores, permitindo uma representação básica de um modelo de um sistema elétrico para o software. A classe *DescricaoProblema* representa um modelo do sistema elétrico de forma mais completa, permitindo uma manipulação dos dados do modelo de forma mais efetiva.

A representação do modelo que esta sendo manipulado pela ferramenta se dá através da classe *ProblemaPrincipal* que faz o controle do modelo com os seus dados de momentos, permitindo, por exemplo, que sejam recuperados os momentos que estão presente no sistema para o determinado modelo, assim como a remoção e a inserção de novos modelos. Para fornecer este tipo de funcionalidade esta classe utiliza classes como *DescricaoProblema* e *DadosMomento*.

Em *DadosMomento* temos os dados do momento, como nome, descrição e período de tempo ao qual ele se refere, porém a principal informação armazenada por esta estrutura diz respeito à lista de elementos com os seus respectivos dados. Um dos motivos pelo quais a classe *ConjuntoExecucao* a acessa.

Através da estrutura de instâncias da classes *ProblemaPrincipal* e dos respectivos *DadosMomento* a classe *ConjuntoExecucao* consegue-se manter um controle das simulações já realizadas, assim como das simulações que estão sendo realizadas, através de instâncias da classe *ControleSimulacao*. A classe *ConjuntoExecucao* também permite que sejam definidos os parâme-

tros de controles como número de iterações e limites de tempo de execução.

Nas estruturas *Simulacao* e *Resultados* são armazenadas as informações da simulação. Em *DadosSimulacao* estão informações como nome, descrição, tempo utilizado para execução, módulo utilizado, *status* final da simulação, entre outros dados. Em *Resultados* estão os resultados numéricos da simulação para cada elemento do sistema elétrico, caso o módulo de cálculo implemente a funcionalidade é possível que os resultados de cada iteração do algoritmo sejam armazenados.

4.4.3 Construção de Gráficos

No pacote *Graficos* estão as funcionalidades capazes de extrair do conjunto de dados aqueles que devem ser utilizados para a construção de gráficos, as classes principais deste pacote podem ser observadas através do diagrama presente na Figura 4.6. O elemento principal é a classe *Grafico* que fornece a abstração necessária para que os gráficos sejam inseridos nas interfaces gráficas com o usuário.

Nesta classe está presente a funcionalidade que permite que o gráfico seja exportado como uma figura no padrão *PNG*¹⁷, assim como a funcionalidade que permite que sejam definidas quais cores serão usadas para desenhar no gráfico a representação dos dados de cada elemento de acordo com a sua posição em relação aos limites operacionais definidos pelo usuário.

Para cada tipo de gráfico foi construído uma classe que deve implementar os métodos de inserção de dados, definição dos limites para os dados que estão sendo analisados e, principalmente, a construção da interface gráfica para o usuário do gráfico definido.

Tem-se no diagrama presente na Figura 4.6 os tipos de gráficos *BarraGraf* que é capaz de desenhar gráficos em barras verticais para qualquer tipo de dados, *LinhaGraf* capaz de desenhar gráficos em linhas para qualquer tipo de dado, *VarGraf* é um gráfico especializado que cria gráficos para observação da variação da tensão de acordo com a variação de carga, *CircularGraf* desenha gráficos circulares para todos os tipos de dados e *BarraGrafHor* responsável pela construção dos gráficos de barras horizontais para qualquer tipo de dado. Os resultados obtidos por estas classes poderão ser observado na Seção 5.1.2.

¹⁷Portable Network Graphics

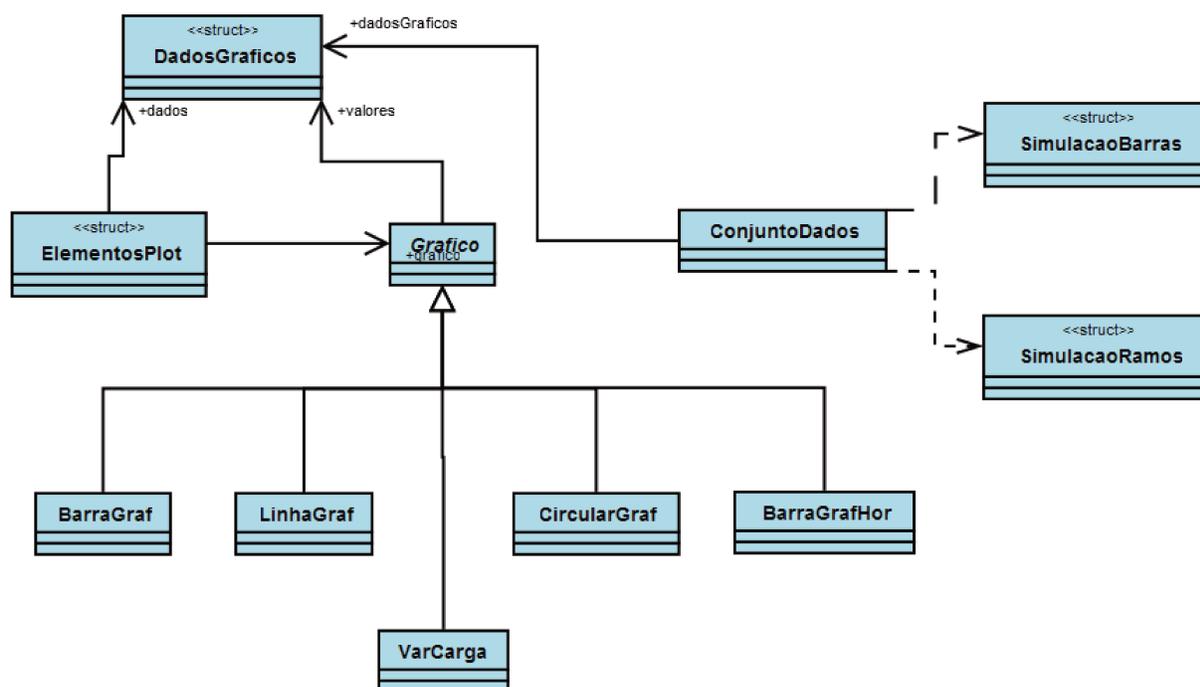


Figura 4.6: Diagrama de classes, visão gráficos

A classe *ConjuntoDados* é responsável por construir as listas que contêm estruturas como *SimulacaoBarras* e *SimulacaoRamos* que fornecem os dados para as classes construtoras de gráficos citadas anteriormente, para que elas possam manipular e obter os dados de forma transparente.

As última duas estruturas do pacote são *DadosGraficos* e *ElementosPlot*. Em *DadosGraficos* temos informações que permitem relacionar *Grafico* com dados necessários para a construção da interface gráfica do usuário. A estrutura *ElementosPlot* permite relacionar os dados do elementos com suas representações na interface gráfica com o usuário, possibilitando, principalmente interações que serão discutidas no tópico 5.1.2.

4.4.4 Elementos do Sistema Elétrico

Neste pacote temos as classes que representam os elementos reais dos sistemas de potência. O diagrama de classes pode ser observado na Figura 4.7. As duas classes principais, *BaseGUI* e *ListaObjetos*, permitem que os elementos assumam o papel adequado de acordo com

quais estruturas do sistemas eles estão se relacionando.

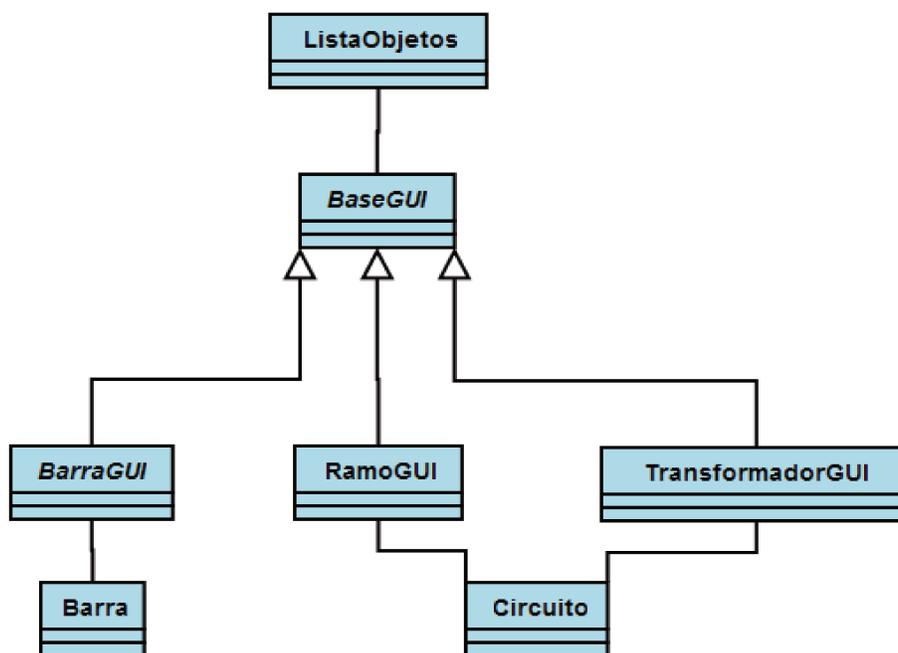


Figura 4.7: Diagrama de classes, visão elementos do Sistema Elétrico

Através de *BaseGUI* são fornecidos os métodos e parâmetros necessários para que cada elemento do sistema elétrico seja visto como um elemento de interface gráfica com o usuário dentro do software. Estas funcionalidades são necessárias para a construção das interfaces gráficas com o usuário, como, por exemplo, a verificação se determinada coordenada da tela esta dentro dos limites da representação gráfica do elemento, a determinação de posicionamentos e movimentações na tela, o tratamento dos sinais emitidos pela interação com o usuário e a inserção de dados que podem ser exibidos junto ao modelo.

A classe *ListaObjetos* fornece uma forma fácil de localizar e manipular as instâncias de todos os elementos que compõem o modelo do sistema elétrico. São mantidos índices baseados em IDs padrões que todo objeto possui, índices de acordo com o tipo de elemento do sistema elétrico que a instância representa, índices baseados nos elementos que possuem contingências ou não. Este classe assim como seus índices são utilizada em praticamente todos os pacotes do sistema de forma direta ou indireta.

As classes *Barra* e *Circuito* tratam os elementos de acordo com a visão de dados do sistema elétrico, assim através destas classes é possível informar ou verificar se o elemento repre-

sentado pelo objeto possui uma contingência ou, por exemplo, qual momento do sistema elétrico os dados do objeto representam.

4.4.5 Interface gráfica

Este pacote contém as classes para as diversas interfaces gráficas com o usuário presentes no software desenvolvido. Devido ao grande número de interfaces imagina-se não ser produtiva a exibição de um diagrama de classes para este pacote. Todas as interfaces possuem no mínimo duas classes, as interfaces com um comportamento mais complexo foram subdivididas, fazendo com que cada parte seja considerada uma subinterface e, assim como para as interfaces, são duas classes para cada subinterface.

Para o projeto dos elementos da interface gráfica com o usuário foi utilizado o padrão de projeto *Model-View-Controller* (MVC), dessa forma procura se isolar a área da lógica do negócio da área de interface gráfica do usuário, facilitando o processo de mudança em ambas as áreas [Lar01].

A primeira classe de cada interface gráfica com o usuário é responsável por construir a interface gráfica em si. Posicionando os elementos, assim como fazendo todo o controle inicial dos sinais emitidos pelo sistema operacional para sinalizar as interações com o sistema operacional e com o usuário.

A segunda classe é responsável por implementar a parte de funcional da interface e tratar os sinais de interação com o usuário de forma efetiva. Dessa forma procuramos isolar a parte gráfica da interface da sua parte funcional, diminuindo ao máximo a interdependência.

4.4.6 Controle de módulos

O último pacote a ser analisado é o controle de módulo inseridos no sistema, que funcionam como alternativas de cálculo. O diagrama de classes composto pela principais classes que compõem este pacote pode ser observada na Figura 4.8.

A classe de controle *Algoritmos* é responsável por fornecer ao sistema a lista de módulos presentes, assim como controlar a inserção de novos módulos, a remoção de módulos já

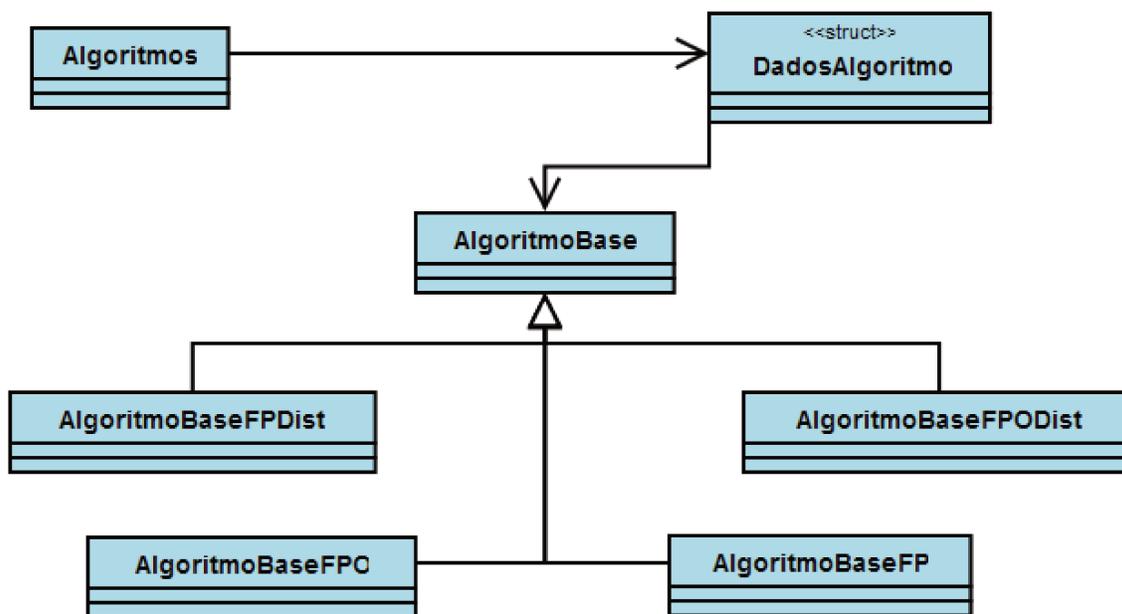


Figura 4.8: Diagrama de classes, visão controle de módulos

existentes e a *linkagem* dinâmica para os módulos anteriormente inseridos, permitindo ao controle de simulação requisitar uma instância de qualquer módulo para a realização das simulações.

Os módulos a serem inseridos no sistema devem derivar de uma das quatro classes: *AlgoritmoBaseFPDist*, *AlgoritmoBaseFPODist*, *AlgoritmoBaseFPO* e *AlgoritmoBaseFP*. Todas estas classes derivam de *AlgoritmoBase*. No protótipo desenvolvido está implementada somente a última classe.

A classe *AlgoritmoBase* é uma classe virtual pura, dessa forma ela obriga que suas classes derivadas só possam ser instanciadas caso implementem certos métodos. Os métodos em questão dizem respeito a uma maneira simples de comunicação entre o sistema e o módulo inserido, permitindo que o sistema sinalize quando a simulação pode ser iniciada e quando deve ser finalizada.

Estas classes também fornecem os métodos que permitem ao módulo acesso aos dados necessários para a simulação, assim como os métodos necessários para informar os resultados obtidos durante as simulações. É feito um controle para evitar que erros que o módulo cometa durante a sua execução comprometam o funcionamento do sistema como um todo, porém nem sempre é possível prevenir que isto ocorra.

4.5 Conclusão

Neste capítulo procurou-se enfatizar os aspectos computacionais do software proposto. Mostrando as tecnologias e os recursos utilizados para a implementação do protótipo do sistema, ressaltando que os recursos e ferramentas utilizadas na implementação e projeto do software proposto são distribuídos, em sua maioria, de acordo com a licença GPL¹⁸. Tal fato acabou por não gerar custo financeiros para aquisição dos recursos de software utilizados no desenvolvimento.

Foram detalhadas como foram feitas as implementações das funcionalidades do sistema, partindo da visão do usuário. O sistema foi mostrado como um conjunto de pacotes de funcionalidades. Cada pacote foi parcialmente detalhado e, de cada pacote, foram descritos em detalhes algumas das principais classes implementadas.

Também foram mostrados detalhes do modelo do banco de dados desenvolvido, isto é, como os dados dos modelos elétricos, das alterações e das simulações são armazenados.

No Capítulo 5 serão demonstradas os resultados dessas implementações, explorando os recursos disponíveis no protótipo construído.

¹⁸Do Inglês: *Gnu Public License* disponível em <http://www.gnu.org/licenses/gpl.html>

Capítulo 5

Descrição das funcionalidades do sistema proposto

O protótipo implementado visa possibilitar a análise de vários aspectos da simulação e análise dos resultados em sistema elétrico de potência, buscando que os objetivos descritos na Seção 3.3 sejam alcançados. Para isso foram construídos módulos gráficos que permitem a visualização dos resultados, a comparação destes resultados com resultados obtidos em simulações de diferentes momentos do sistema, a alteração de parâmetros de elementos do sistema, assim como também, da topologia da rede.

Uma funcionalidade que existe, para auxiliar operadores e engenheiros, é a exportação dos gráficos gerados para um formato de imagem, o que possibilita fácil manipulação, permitindo a inserção destas figuras em documentos e relatórios.

Foi implementado o controle dos momentos/configurações do sistema, o que permite que o usuário altere os dados, ou as configurações do sistema, realizando simulações e salvando, ou não, os resultados no banco de dados.

O sistema usa o algoritmo desenvolvido em [Bau05] como padrão para o cálculo do fluxo de potência, porém existe a possibilidade da inserção de novos módulos de cálculo, sem a necessidade de recompilar, ou mesmo, reiniciar o sistema.

Foi implementada, também, uma funcionalidade que permite ao usuário a análise da

sensibilidade do modelo elétrico estudado, possibilitando a inserção de contingências em circuitos, ou até mesmo em barras, o que acaba por caracterizar, na maioria dos casos, em uma contingência muito severa ao sistema, impossibilitando a realização de simulações. É permitido também a inserção de novas barras e circuitos ao sistema, ficando a cargo do usuário a definição dos dados destes elementos para que as simulações realizadas nesses casos tornem-se viáveis computacionalmente.

Neste capítulo serão descritas as diversas interfaces e funcionalidades que compõem o sistema. Na Seção 5.1 serão apresentadas as funcionalidades do sistema de acordo com o agrupamento de janelas, mostrando a visualização do modelo, dos gráficos, dos dados do sistema e o controle de simulações. Na Seção 5.2 será apresentada a organização das funcionalidades descritas anteriormente, mas sem o uso do agrupamento de janelas. Na Seção 5.3 será apresentada a funcionalidade que permite que novas rotinas de cálculo para o fluxo de potência sejam inseridas no sistema. Na Seção 5.4 será apresentada a conclusão dos resultados obtidos com o protótipo implementado.

5.1 Agrupamento de dados para acesso às funcionalidades do sistema

Esta interface com o usuário permite que todas as representações do sistema de potência que esta sendo analisado sejam agrupadas, buscando assim facilitar a navegação e o controle, por parte do usuário, dos dados que estão sendo manipulados. Inicialmente é exibido o diagrama que representa a primeira barra, junto com suas vizinhas, do sistema elétrico que esta sendo analisado. A primeira barra é definida de acordo com a ordem de inserção das barras no banco de dados.

Esta interface foi construída baseada no conceito TDI [Kum05], do inglês *Tabbed Document Interface*, que permite o agrupamento de interfaces que possuem uma forte relação entre elas, neste caso as interfaces que permitem a visualização dos dados do sistema elétrico analisado, a alteração destes dados, o controle das simulações sobre eles e a análise destes dados.

Computacionalmente, este tipo de abordagem possui a vantagem de ter um *footprint*¹ de memória menor levando-se em consideração que para cada uma das funcionalidades uma, ou

¹Quantidade de memória utilizada pelo sistema

mais, janela seriam abertas pelo sistema caso não fosse adotado o conceito TDI.

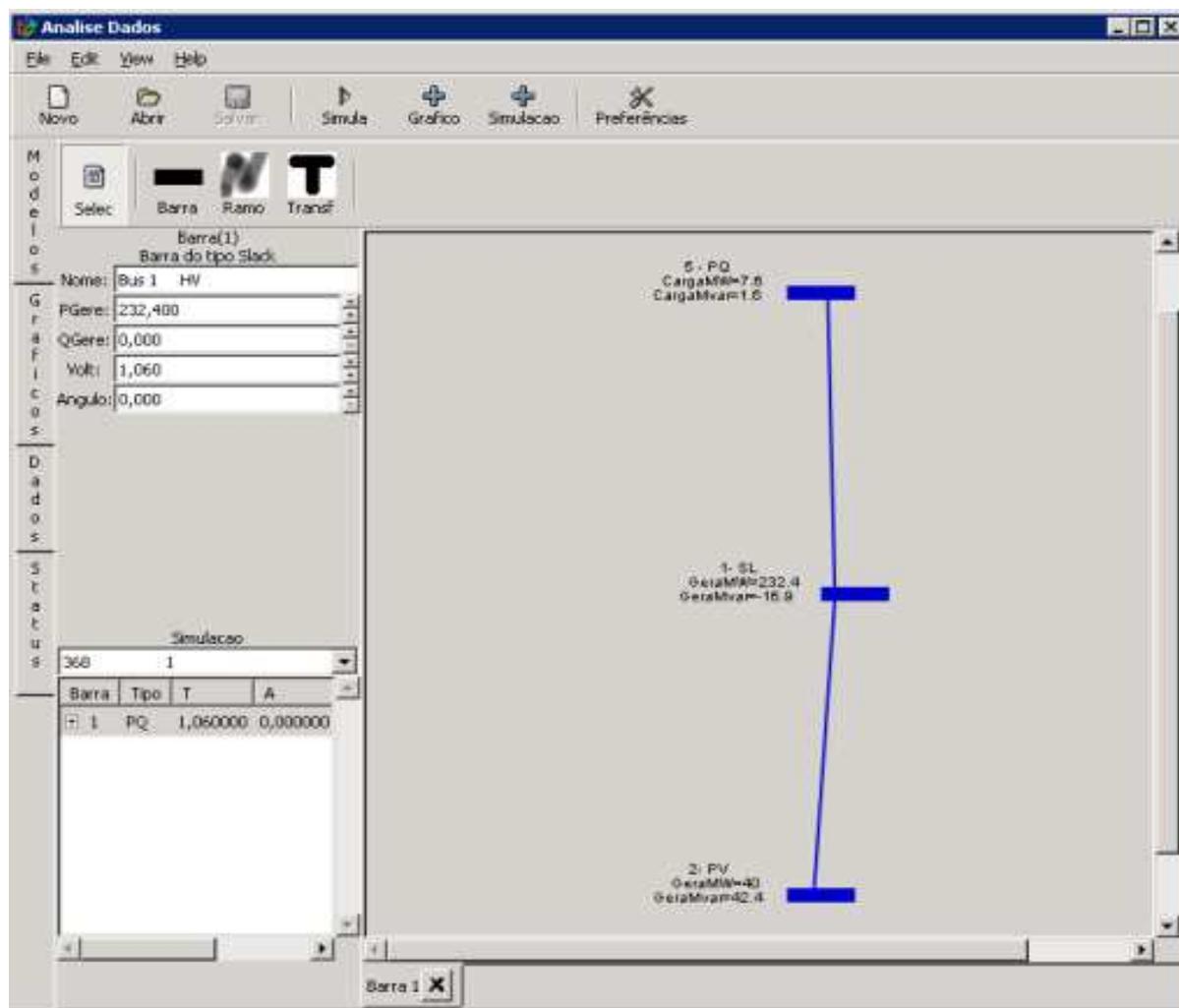


Figura 5.1: Interface para Agrupamento de Dados do Modelo

A abordagem TDI tem como clara desvantagem o fato de que toda aba estará sempre maximizada dentro da janela principal, permitindo assim que somente uma aba seja visualizada por vez. Por esta razão existe a possibilidade do usuário utilizar janelas separadas para cada tipo de análise ou manipulação de dados. Este tipo de construção é conhecido como SDI, do inglês *Single Document Interface*, onde uma janela independente é aberta em cada situação.

A principal desvantagem da abordagem SDI [Kum05] para o sistema implementado é que facilmente o número de janelas abertas cresce, podendo tornar a navegação complexa e

confusa. A tela inicial do software foi construída baseada na tecnologia SDI e será mostrada na Seção 5.2. Através das interfaces com o usuários descritas nessa seção é que se é possível iniciar a interface mostrada na Figura 5.1 e que se utiliza da abordagem TDI.

Com pode ser observado na Figura 5.1, no interior desta janela foram construídas quatro abas principais. A primeira onde é exibido o diagrama do sistema elétrico, a segunda onde são manipulados os gráficos sobre as simulações realizadas, a terceira aba que permite a visualização tabular dos dados e a última aba que permite o controle das simulações do modelo que esta sendo estudado.

Como mostrado na Figura 3.3 presente na Seção 3.3, os passos da estratégia de solução adotada no software proposto são:

1. O primeiro passo são as interfaces interativas, representadas pelo conjunto de funcionalidades inseridas na interface com o usuário mostrada através da Figura 5.1;
2. O segundo passo da estratégia diz respeito à especificação de uma solução inicial para o modelo elétrico que esta sendo analisado, isto pode ser obtido através das funcionalidades descritas na Seção 5.1.1 e 5.2, que se referem à inserção de um modelo de um sistema elétrico e da alteração dos dados inseridos;
3. O terceiro passo diz respeito à solução do fluxo de potência. As interfaces com o usuário que permitem essa funcionalidade serão descrita na Seção 5.1.4 onde será apresentado os recursos necessários para o controle das simulações;
4. O quarto passo da estratégia consiste na análise da solução obtida no passo anterior e será realizada pelas funcionalidades descritas na Seção 5.1.1, 5.1.2 e 5.1.3;
5. O quinto passo consiste na proposta de uma nova solução baseado nas análises feitas no passo anterior, também feito através das funcionalidades da Seção 5.1.2 e 5.1.1;
6. O sexto, e último, passo da estratégia proposta consiste no armazenamento da solução no banco de dados, pode ser feito através das funcionalidades apresentadas na Seção 5.1.4;

5.1.1 Visualização do modelo

Esta aba permite que o modelo unifilar do sistema elétrico que está sendo estudado seja manipulado. O usuário possui quatro grupos de funcionalidades, conforme Figura 5.2.

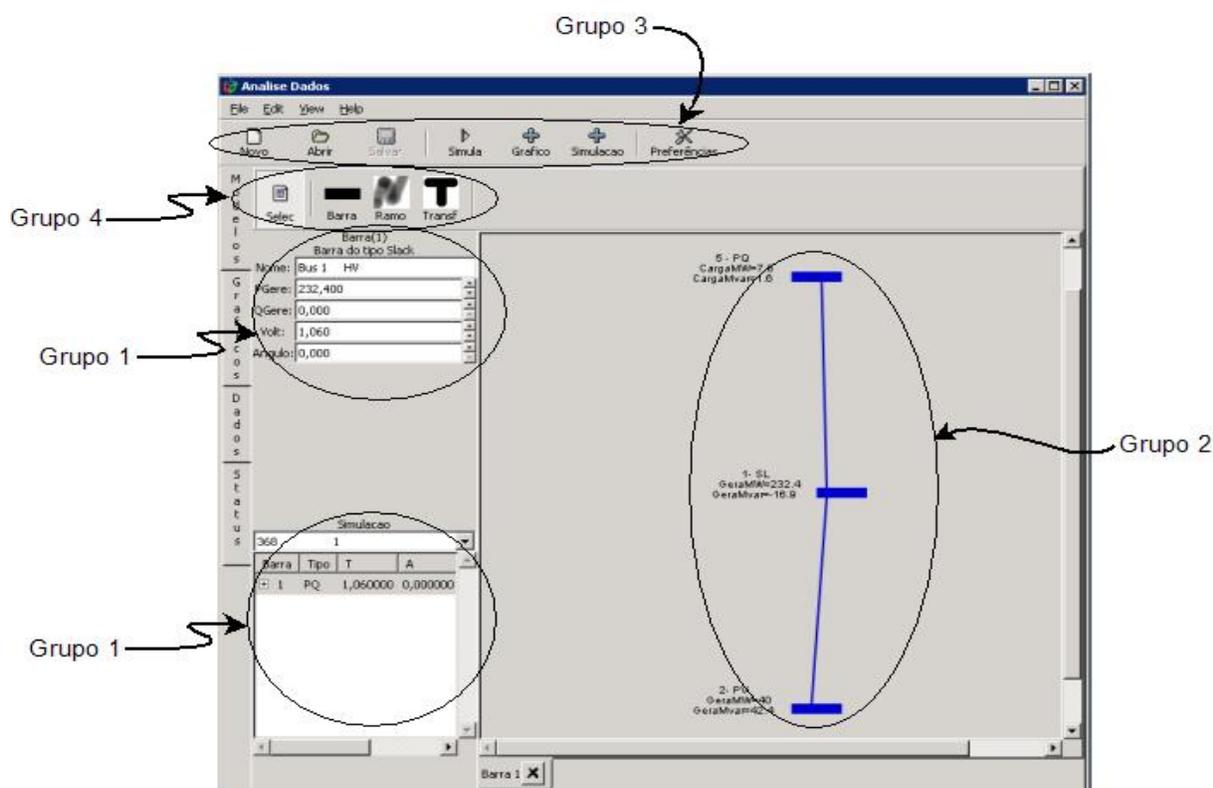


Figura 5.2: Interface para Agrupamento de Dados do Modelo, com grupos destacados

No primeiro grupo é possível manipular os dados dos elementos do sistema elétrico de potência que foi selecionado. Os dados informados devem variar de acordo com o tipo de elemento, assim quando uma barra PQ é selecionada são exibidos os dados relevantes a este tipo de elemento do sistema e, conseqüentemente, são diferentes dos dados exibidos quando uma barra PV é selecionada. Há também, neste grupo, a possibilidade de selecionar a simulação que se deseja colocar em destaque, assim serão exibidos os dados desta simulação junto as representações das barras exibidas no grupo 2, conforme Figura 5.2.

A barra no centro do desenho, como pode ser observada no grupo 2 da Figura 5.2, é

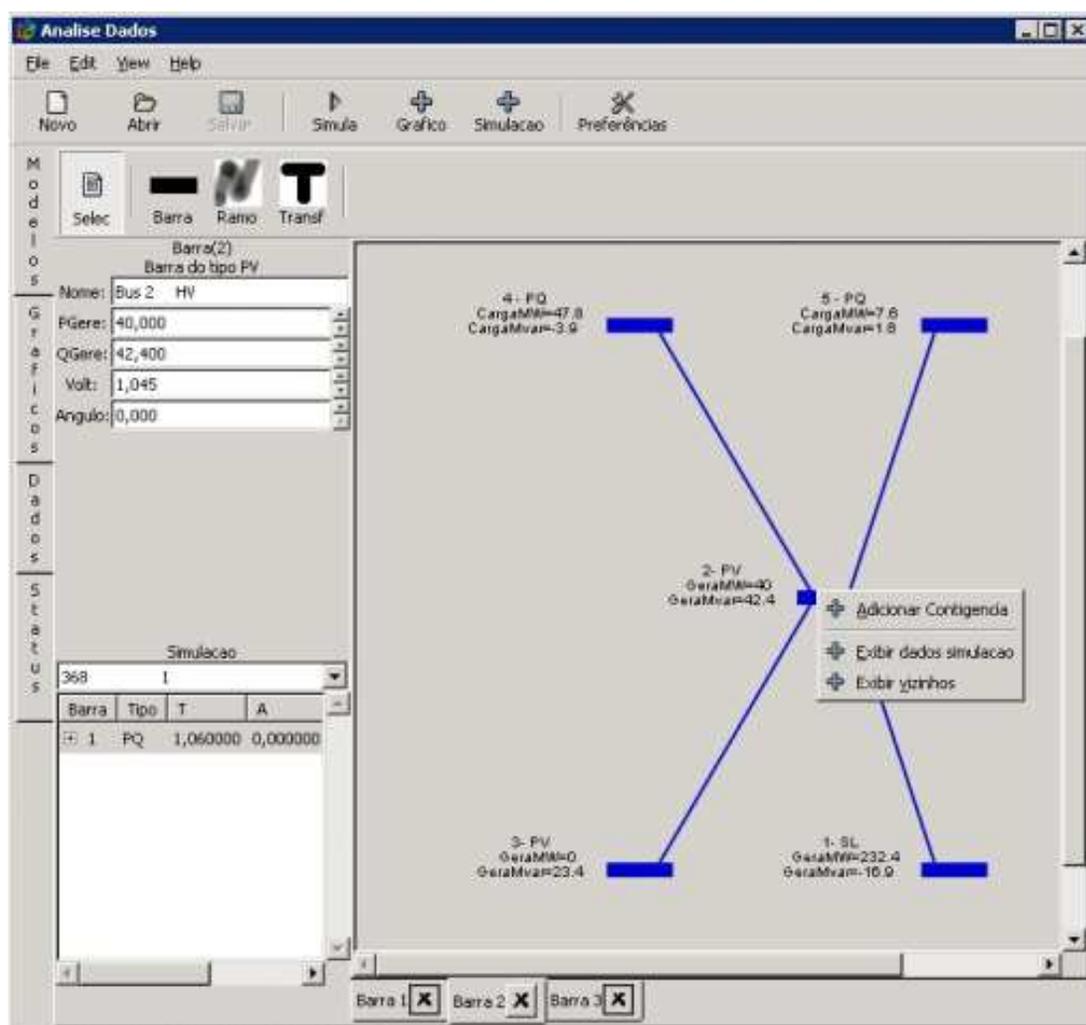


Figura 5.3: Interface para Exibição de Barras Vizinhas

considerada a barra principal da representação, sendo exibidas as suas vizinhas, permitindo dessa forma que o usuário explore as vizinhanças das barras, por exemplo, quanto estiver procurando as barras PV vizinhas de uma barra PQ com tensão infactível. A barra principal dá nome à aba. É possível verificar a vizinhança de qualquer barra, solicitando ao sistema a sua exibição, conforme Figura mostrado na 5.3.

Junto à representação gráfica de cada barra é possível exibir dados do modelo e resultados da simulação seleccionada através da caixa de seleção do grupo 1. É também possível exibir as diferenças entre a configuração seleccionada e a configuração original, bem como as diferenças



Figura 5.4: Interface para Configuração de Campos

nos resultados da simulações sobre estas configurações. O conjunto de informações exibidas junto à representação gráfica de cada barra pode ser específico para cada tipo.

É possível acessar a lista de dados a serem exibidos junto à representação gráfica através das configurações presentes no grupo 3 da Figura 5.2. Como pode ser observado na Figura 5.4 são fornecidas diversas opções de configurações que visam facilitar a análise e a visualização dos resultados obtidos nas simulações.

Além da lista de campos a serem exibidos, que podem ser selecionados conforme exemplifica a Figura 5.4, é possível selecionar as cores, conforme Figura 5.5, que irão destacar os elementos de acordo com os limites operacionais selecionados, conforme mostrado na Figura 5.6. Nesta aba também é possível selecionar quais serão os dados da simulação que serão usados como referência para a aplicação do esquema de cores. Neste caso também é possível escolher dados diferentes, de acordo com a natureza da barra.

A seleção de cores permite que seja definido um gradiente de cores para destacar os

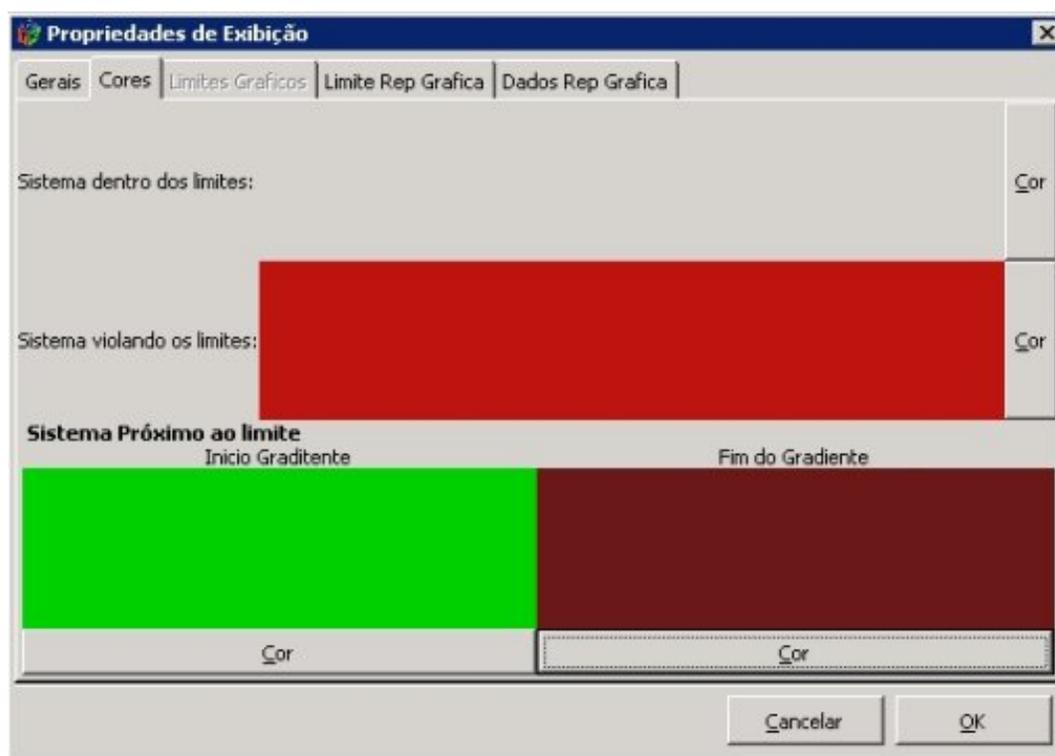


Figura 5.5: Interface para Configuração de Cores

elementos do sistema que não violaram, porém que se encontram a uma valor muito próximo, definido pelo usuário, de violar o limite operacional superior ou inferior, também definido pelo usuário, para os dados que estão sendo analisados.

Com esse mecanismos procura-se evitar que elementos muito próximos, mas que ainda não violaram um limite de segurança do sistema, não sejam analisados de forma correta pelos usuários do sistema.

As definições das cores e dos limites descritos anteriormente não influenciam somente a representação gráfica dos elementos, mas também o gráfico de comparação com outras simulações que pode ser acessado através do grupo 3. Através da Figura 5.7 é possível observar a exibição dos gráficos de comparação junto com a representação gráfica dos elementos do sistema elétrico.

Através das funcionalidades do grupo 3 também é possível inserir, ou retirar, novas simulações dos gráficos que estão sendo exibidos junto às barras. Através das funcionalidades que permitem o acesso e a alteração dos dados do modelo (grupo 1), da visualização da configuração

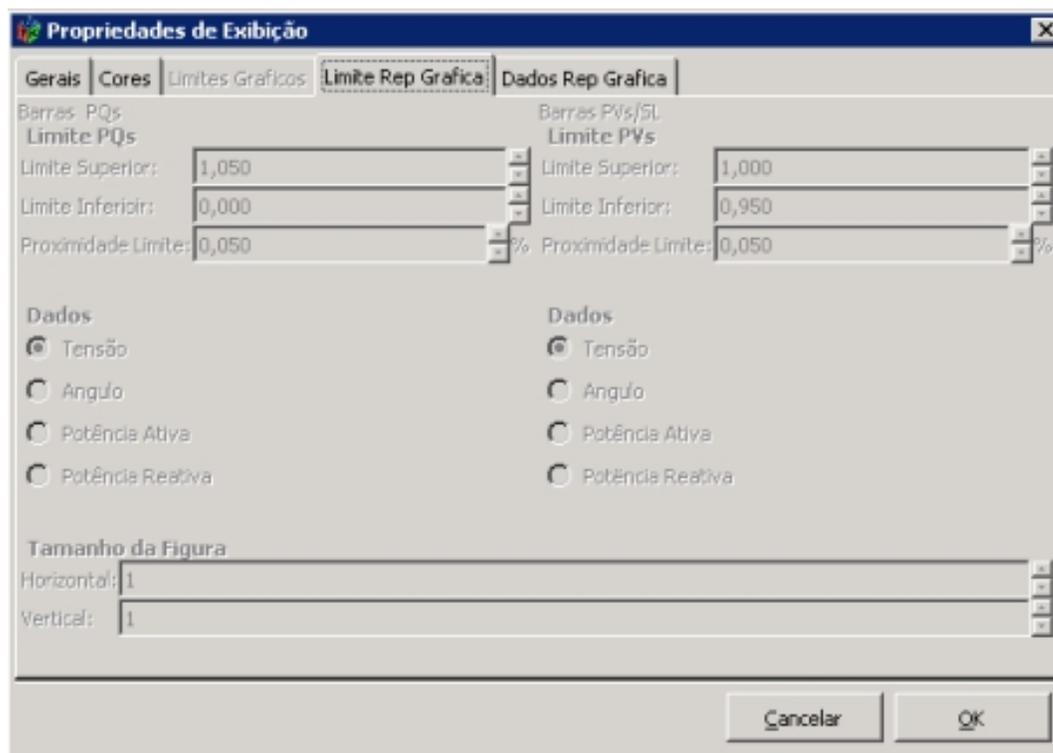


Figura 5.6: Interface para Configuração de Limites

atual do sistema e das diferenças em relação à configuração base e às outras configurações do sistema (grupo 2), assim como os ajustes possíveis (grupo 3) dos elementos em exibição, procura-se dar a possibilidade ao operador de visualizar e verificar o estado atual e como as alterações impostas alteram o sistema de potência em relação às configurações já simulados.

Através da interface que está sendo analisada também é possível a inserção de contingências no modelo. Estas contingências podem ser inseridas ou retiradas, conforme exemplifica a Figura 5.8. É importante ressaltar que o sistema não impede que sejam inseridas contingências em todos os ramos de uma determinada barra, nestes casos a barra é excluída da simulação, podendo esta contingência extremamente severa comprometer a capacidade de convergência do algoritmo utilizado.

No quarto e último grupo de elementos visuais estão presentes aqueles que possibilitam complementar a análise de sensibilidade do modelo, permitindo a inserção de novos elementos elétricos. Para inserir um novo ramo, ou um novo transformador, basta selecionar o novo elemento

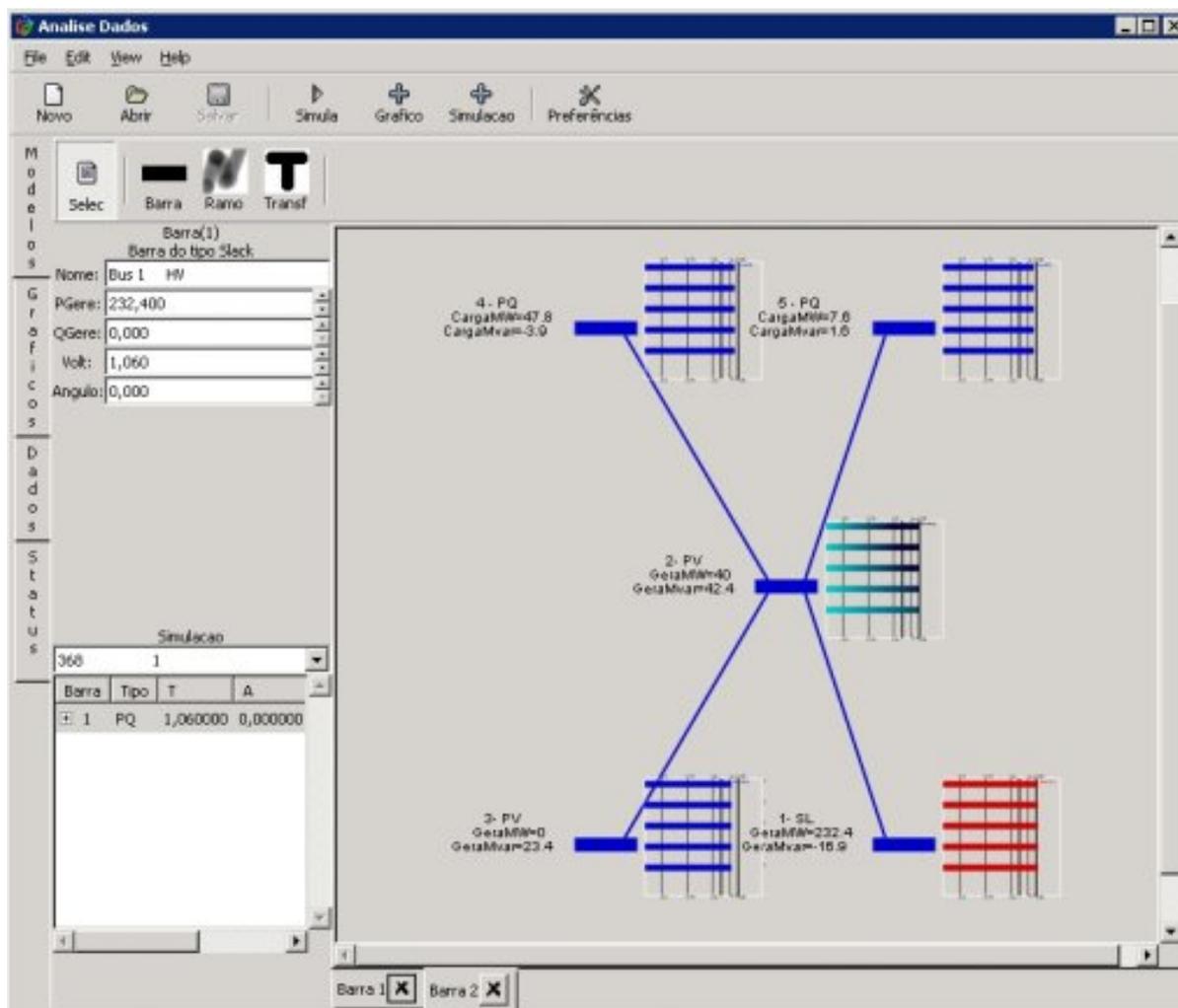


Figura 5.7: Interface com Exibição de Gráficos juntos às Barras

desejado na barra de elementos exibidos pelo grupo 4, clicar sobre a barra inicial e sobre a final e o novo circuito será inserido ligando estas duas barras. Não é feito qualquer tipo de verificação da semântica deste tipo de ligação.

A inserção de uma nova barra se dá de maneira análoga ao que foi descrito para os ramos e transformadores, porém neste caso basta clicar uma vez na representação gráfica do modelo para que a barra seja inserida. Durante a inserção da barra é necessário informar a natureza da barra, ou seja, se é de geração ou de carga.

Sobre nenhum novo elemento inserido no sistema é feito qualquer tipo de verificação

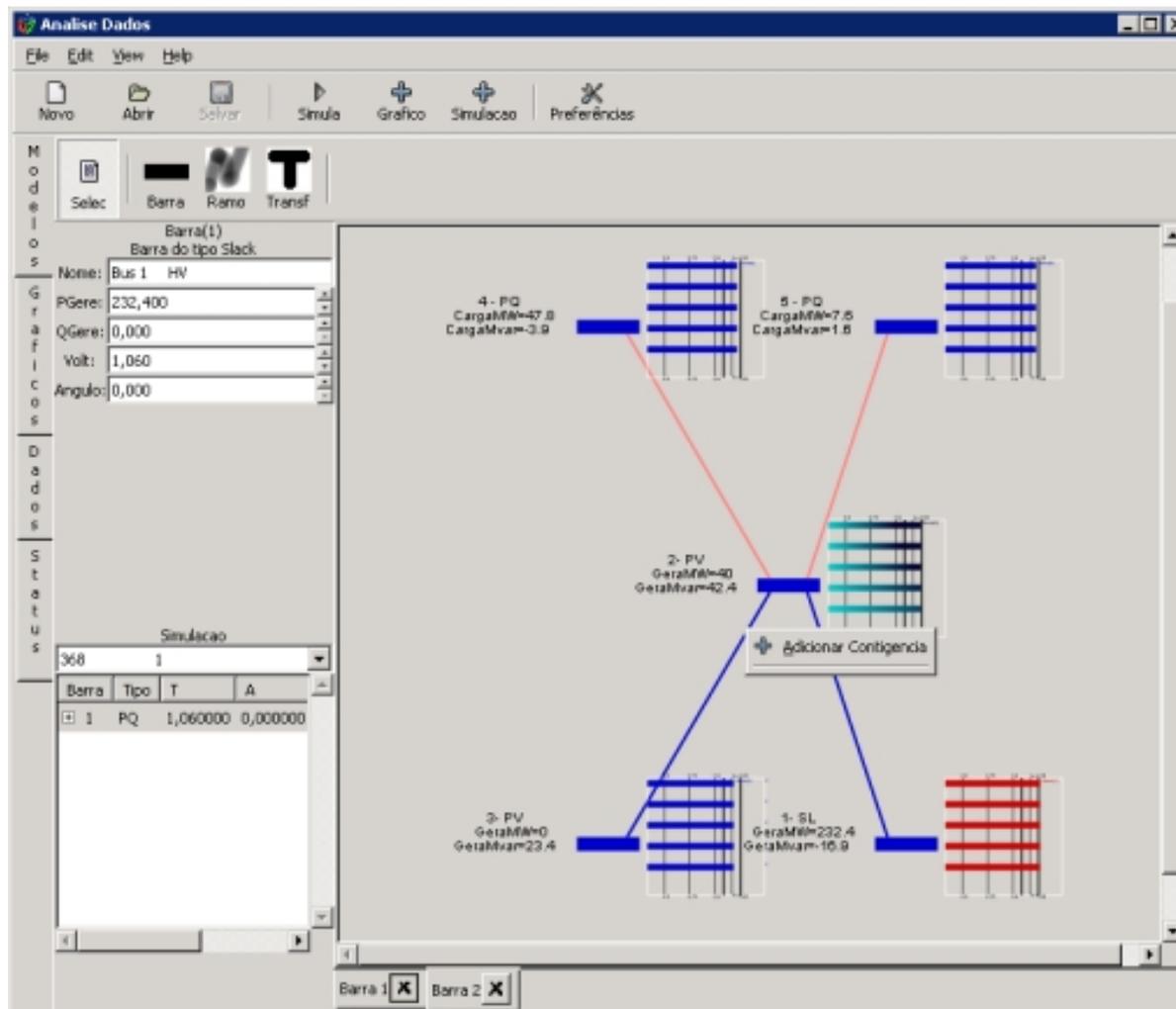


Figura 5.8: Interface para Controle de Contingências

no que tangem aos dados elétricos deste elemento. O sistema basicamente ajusta todos os dados para um valor *default*, fica a cargo do usuário fornecer dados aos novos elementos de forma que a convergência do modelo seja viável. Um exemplo de novos elementos inseridos pode ser observado na Figura 5.9.

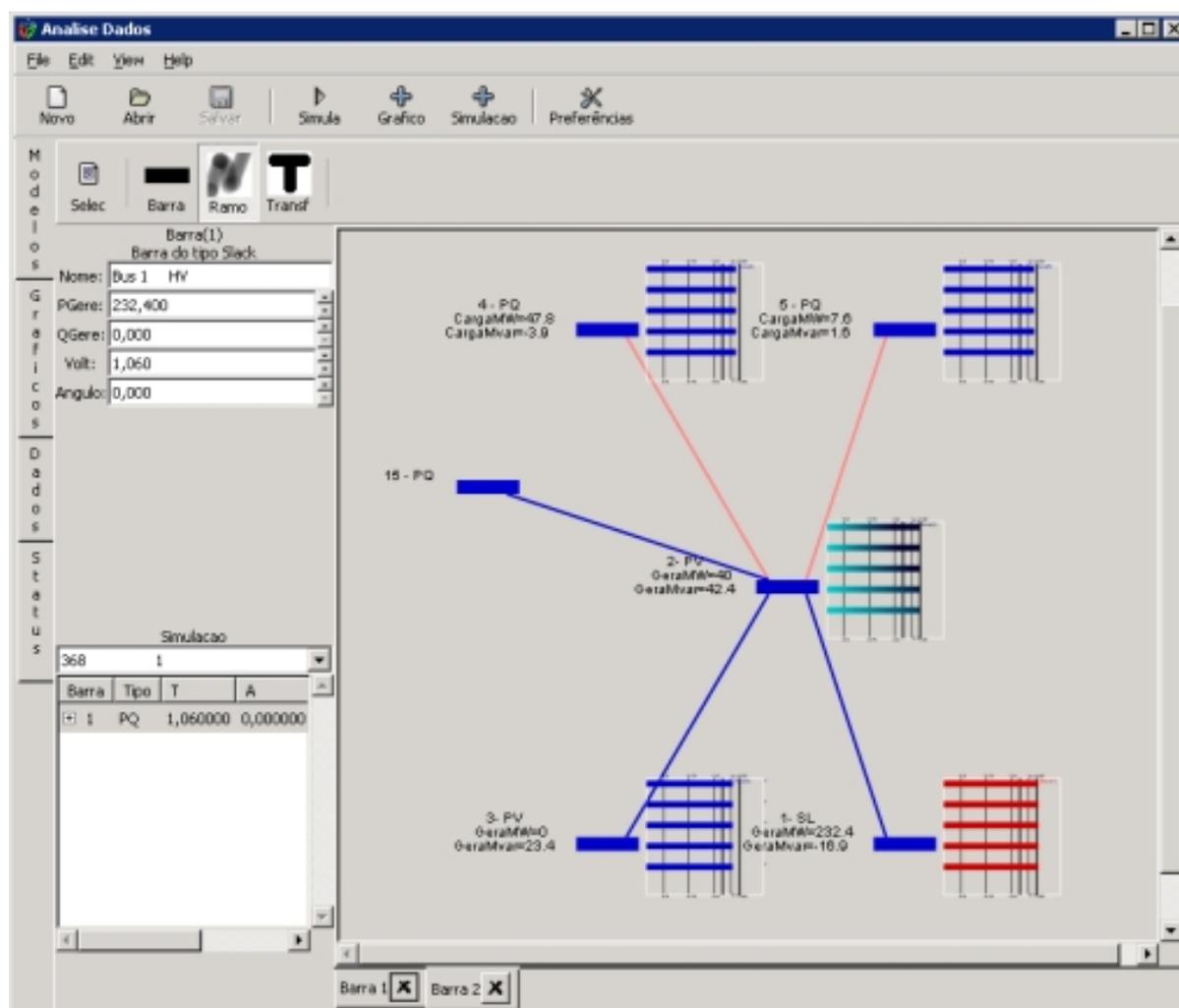


Figura 5.9: Interface para Inserção de Novos Elementos

5.1.2 Visualização dos gráficos

Esta funcionalidade também foi implementada de acordo com o conceito TDI. Sendo possível criar vários gráficos dentro de uma mesma janela de análise. A aba de exibição dos gráficos pode ser observada na Figura 5.10.

O esquema de cores segue o mesmo padrão definido anteriormente na Seção 5.1.1 para a visualização do modelo. O usuário pode definir uma cor para identificar os elementos que se encontram dentro do limite pré-estabelecido para o tipo de dado da rede elétrica que esta sendo

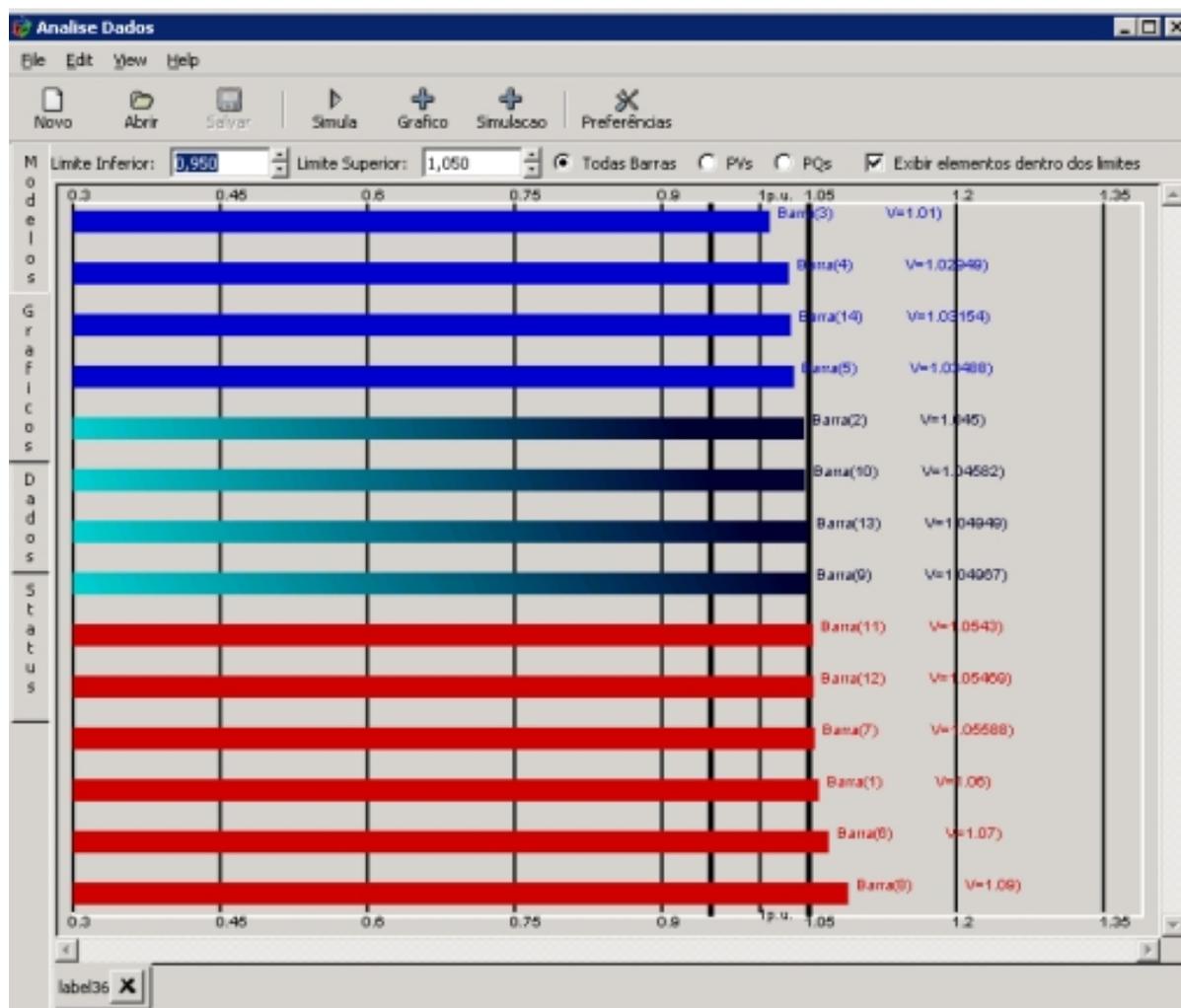


Figura 5.10: Interface para Exibição de Gráficos

exibido, uma cor para os elementos do sistema elétrico que violam o limite e um gradiente de cores para os elementos que se encontram próximos de romper o limite pré-estabelecido.

Como pode ser observado na barra de ferramentas da Figura 5.10 é dada a possibilidade para que o usuário altere os valores de maneira interativa para os limites inferior e superior dos dados que estão sendo exibidos, fazendo com que o sistema redesenhe imediatamente o gráfico, verificando o estado dos elementos com os novos limites. O usuário pode também definir qual a classe de barras serão exibidos, decidindo se deseja ver as barras PQ, PV ou, nos casos em que isto for interessante para a análise, todas as barras. A última opção permite que o usuário determine a

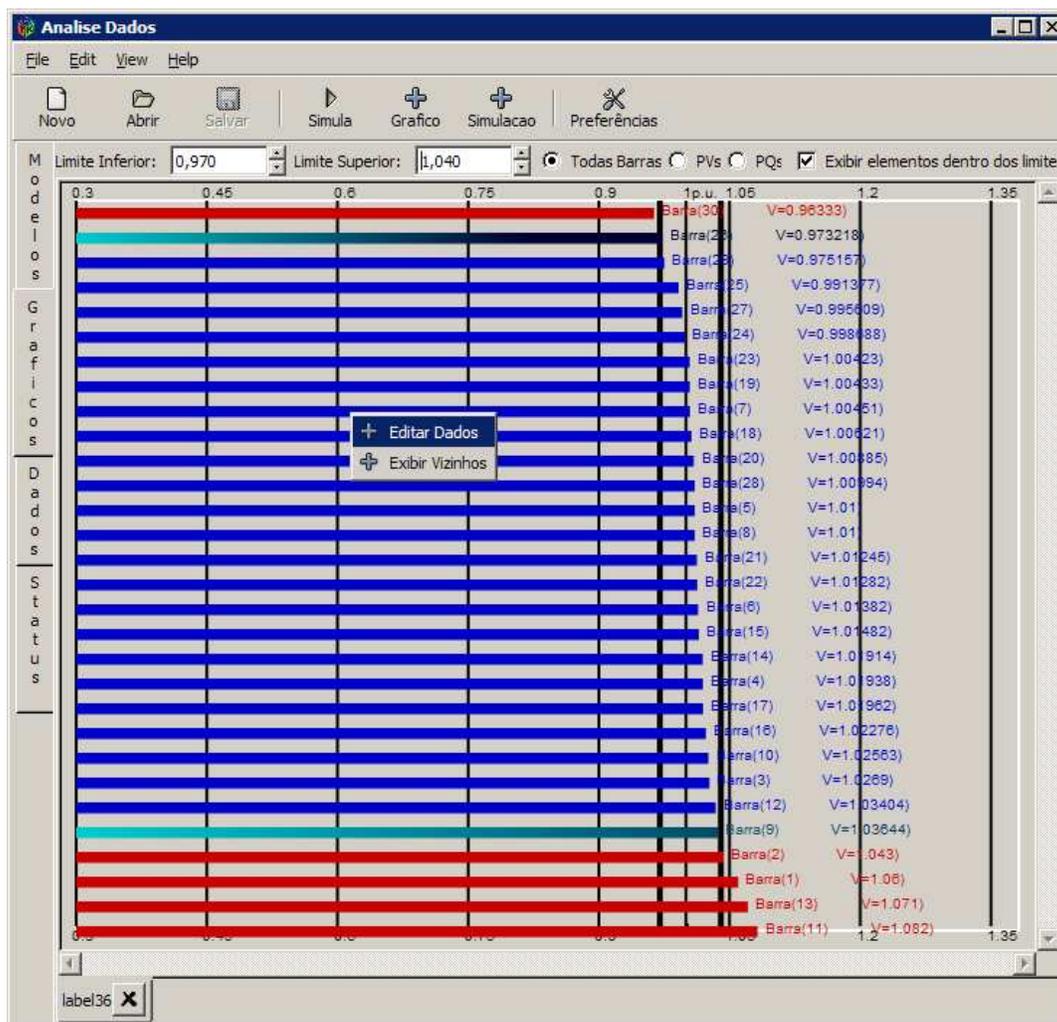


Figura 5.11: Interface com Funcionalidades para Gráficos

exibição, ou não, das barras que estão respeitando os limites para os dados sob análise.

Observando os resultados das simulações no gráfico o usuário pode desejar exibir a vizinhança de uma determinada barra buscando, assim, uma melhor capacidade de analisar o estado do sistema, ou ainda que sejam exibidos os dados relativos àquela barra. Isto pode ser feito como indica a Figura 5.11.

O usuário pode construir gráficos em formas de barras horizontais, como os da Figura 5.10, para todos os tipos de dados obtidos com a simulação do sistema, além destes, é possível a construção de gráficos circulares, conforme Figura 5.12.

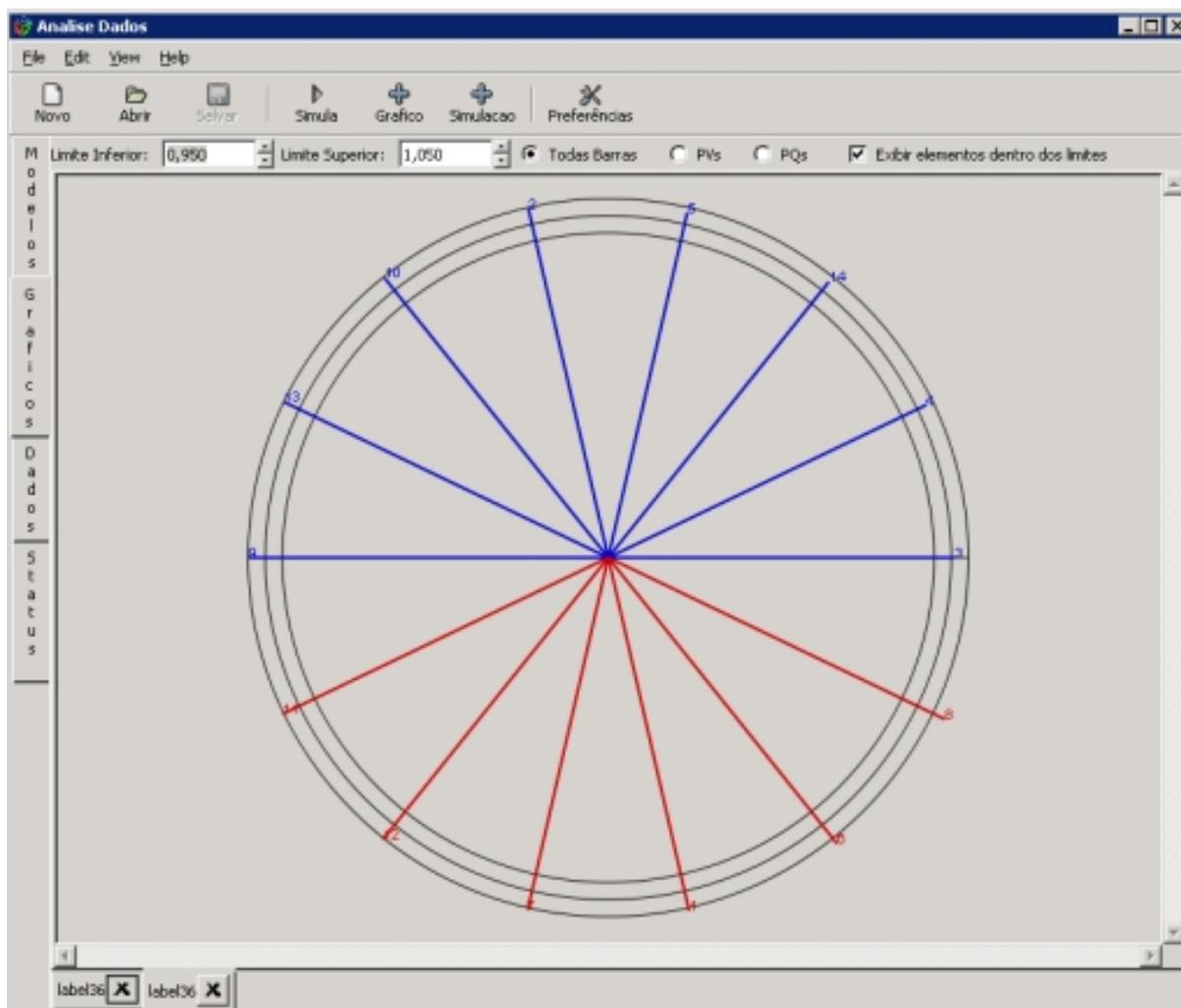


Figura 5.12: Interface para Exibição de Gráficos em Círculos

Existem no sistema dois tipos de gráficos pré-definidos, o primeiro é destinado à análise do reativo, como pode ser observado na Figura 5.13, onde os valores dos resultados são normalizados para que fiquem no intervalo $[0,1]$ quando dentro dos limites definidos pelo usuário. O segundo gráfico permite a análise da variação de tensão de acordo com a variação de carga da barra, dessa forma o usuário seleciona um conjunto de simulações onde haja alterações na carga da barra e verifica como a tensão nesta barra é influenciada.

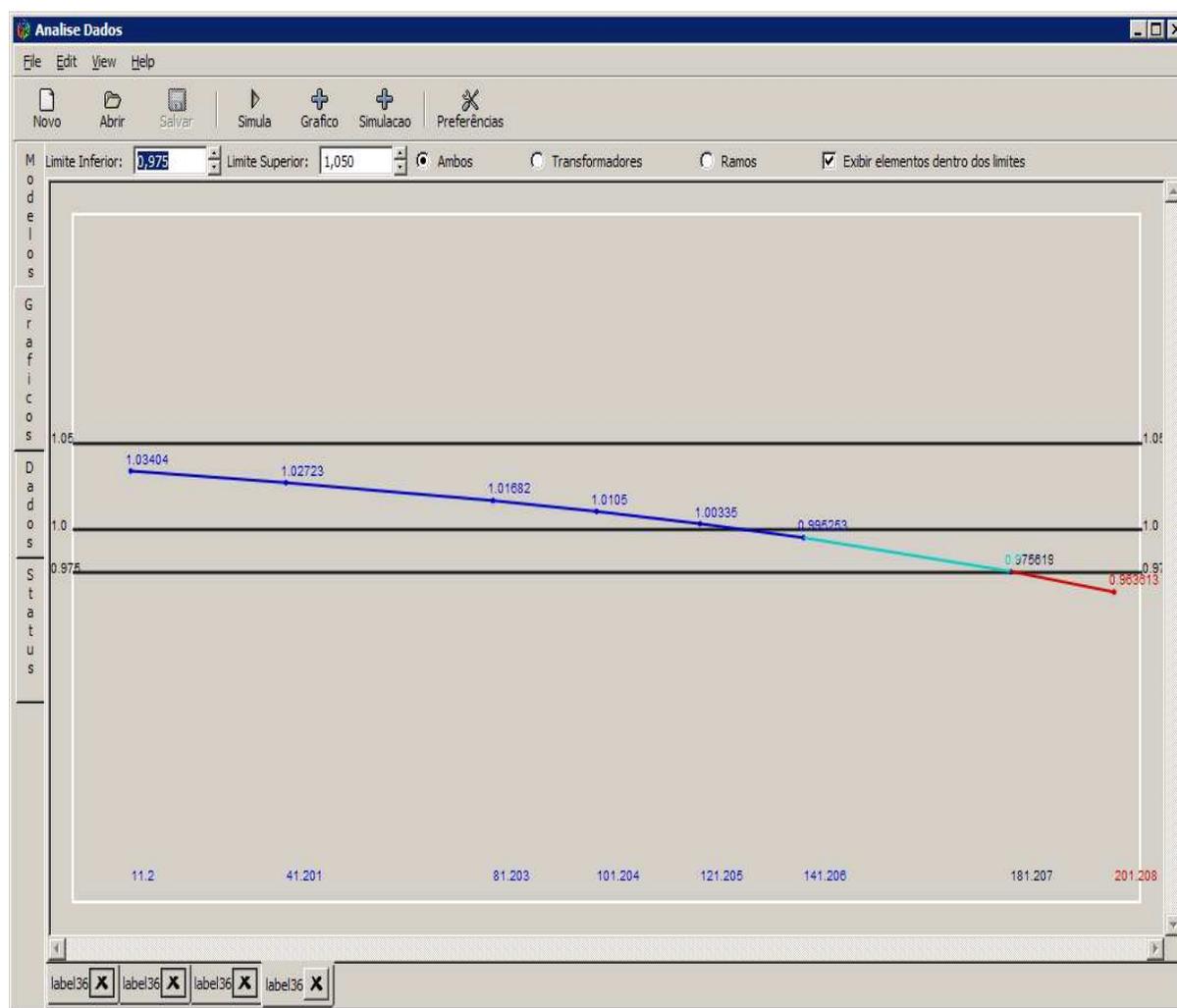
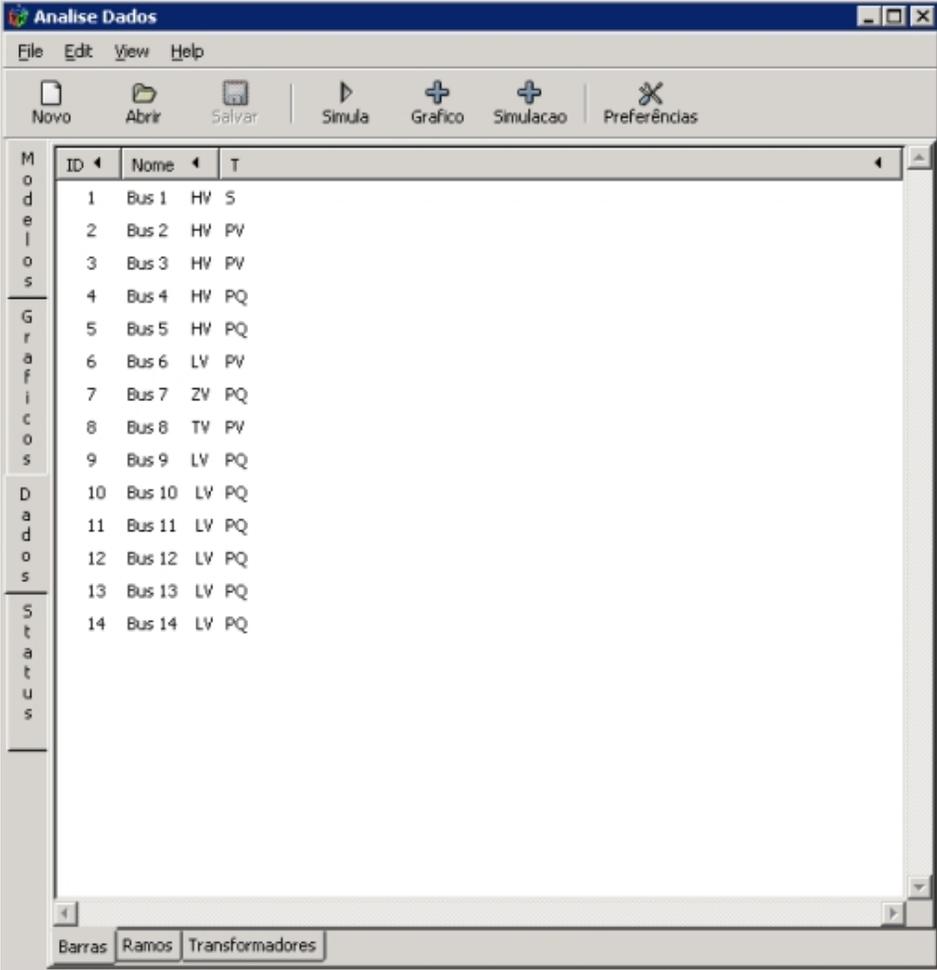


Figura 5.13: Interface para Exibição de Potência Reativa

5.1.3 Visualização dos dados sistema

Esta funcionalidade permite ao usuário visualizar numericamente e manipular os dados do sistema de forma tabular. O conjunto de dados a serem exibidos para cada barra do sistema pode ser configurado conforme mostrado na Figura 5.14.

O sistema permite que o usuário ordene as barras de acordo com um dado específico. Além dos dados das barras, podem ser exibidos os dados dos ramos e dos transformadores que compõem o modelo do sistema elétrico que está sendo analisado, sobre os quais também podem



The screenshot shows a software window titled 'Analise Dados'. The menu bar includes 'File', 'Edit', 'View', and 'Help'. The toolbar contains icons for 'Novo', 'Abrir', 'Salvar', 'Simula', 'Grafico', 'Simulacao', and 'Preferencias'. On the left side, there is a vertical menu with options: 'Modelos', 'Gráficos', 'Dados', and 'Status'. The main area displays a table with the following data:

ID	Nome	T
1	Bus 1	HV S
2	Bus 2	HV PV
3	Bus 3	HV PV
4	Bus 4	HV PQ
5	Bus 5	HV PQ
6	Bus 6	LV PV
7	Bus 7	ZV PQ
8	Bus 8	TV PV
9	Bus 9	LV PQ
10	Bus 10	LV PQ
11	Bus 11	LV PQ
12	Bus 12	LV PQ
13	Bus 13	LV PQ
14	Bus 14	LV PQ

At the bottom of the window, there are three buttons: 'Barras', 'Ramos', and 'Transformadores'.

Figura 5.14: Interface para Exibição Tabular dos Dados

ser realizadas funções de ordenação.

5.1.4 Controle da simulação

O sistema implementado possui uma funcionalidade que permite que sejam definidos parâmetros para o controle das simulações, pode-se citar: o número máximo de iterações desejadas e o tempo máximo de execução do algoritmo. O controle do número de iterações deve ser implementado no módulo de cálculo inserido no sistema.

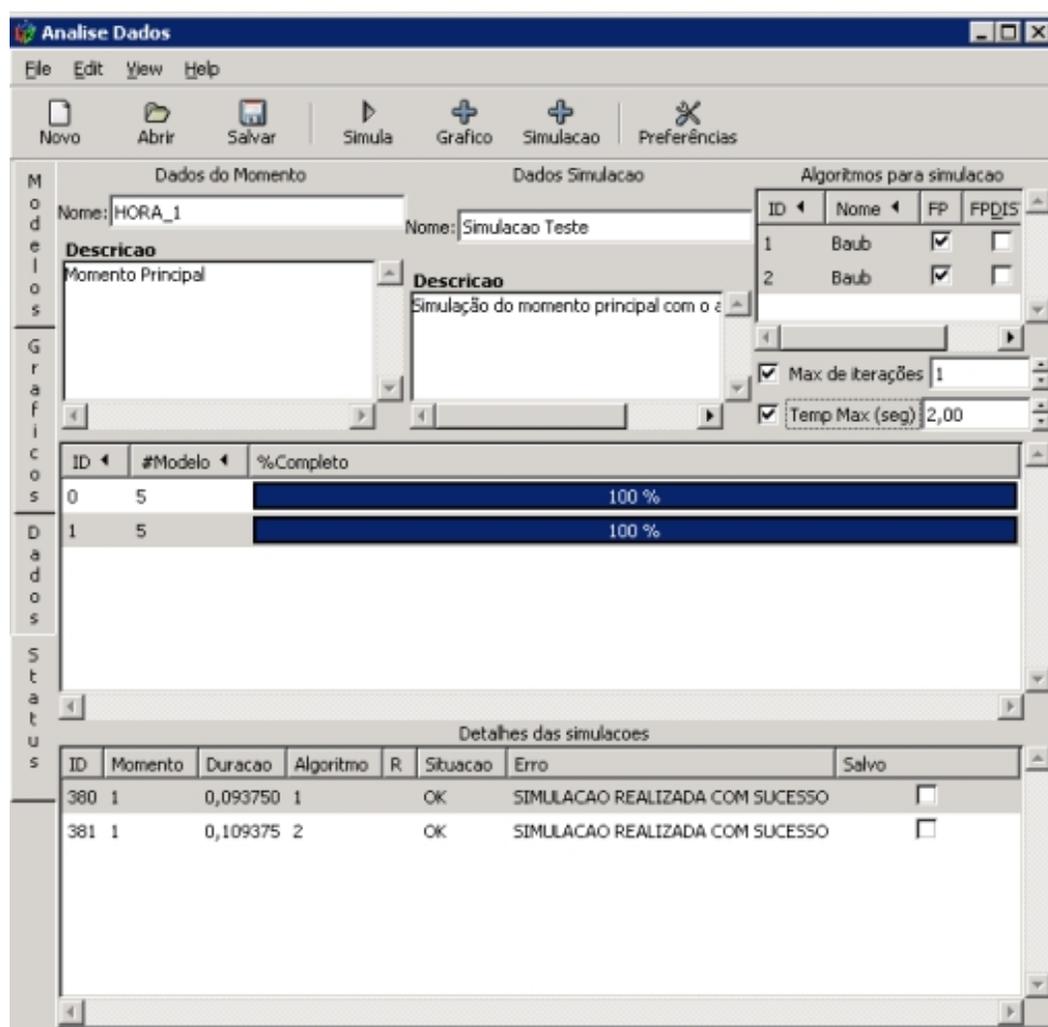


Figura 5.15: Interface para Controle de Simulação

O tempo máximo de execução é controlado pelo sistema e, ao ser alcançado, é enviada uma mensagem ao módulo de cálculo, solicitando que ele finalize a sua execução. É permitido ao módulo salvar o estado em que se encontrava quando foi interrompido.

Esta interface com o usuário também permite a seleção dos algoritmos que serão utilizados para a simulação e o acompanhamento da evolução do processo de simulação, através da interface representada na Figura 5.15. Dessa forma procura-se seguir uma regra básica no desenvolvimento de interfaces que é manter o usuário informado do estado que o sistema se encontra e do que esta ocorrendo [MWKC06].

Ao terminar a simulação é possível verificar o seu estado final, observando se houve, ou não, algum tipo de erro. As simulações, ao terminarem, são automaticamente inseridas na lista de simulações que o usuário tem disponível para efetuar as suas análises nas outras interfaces do sistema.

É neste momento que o usuário pode nomear e inserir comentários, conforme se observa na Figura 5.15, sobre as simulações e os cenários sobre os quais as simulações foram realizadas. É nesta interface também que fica disponível a possibilidade de se salvar a simulação na base de dados, caso seja de interesse do usuário.

5.2 Navegação autônoma pelo sistema

Todos os recursos descritos anteriormente na Seção 5.1 podem ser acessados sem a utilização do conceito TDI. O acesso se dá através da interface principal do sistema que pode ser observada na Figura 5.16.

Através do menu *File* são fornecidas três funcionalidades: a primeira torna possível iniciar a interface com o usuário de acordo com o conceito TDI, conforme descrito na Seção 5.1, a segunda funcionalidade diz respeito à inserção de novos algoritmos, o que será apresentado na Seção 5.3, e a terceira funcionalidade diz respeito a inserção de novos modelos de redes elétricas.

Quando o usuário utiliza as funcionalidades do software, novas janelas que não estão mais agrupadas, são criadas, sendo possível navegar por elas através da barra de tarefas ou outros mecanismos de navegação interjanelas fornecidas pelo sistema operacional, conforme Figura 5.17.

A interface que permite a visualização de gráficos funciona de maneira semelhante ao descrito no tópico 5.1.2 e pode ser acessada através do botão "*Perfis de Potência*" que pode ser observado na Figura 5.16.

O controle de simulações, que pode ser acessado pelo botão "*Controle Simulações*" fornece as mesmas funcionalidades descritas no tópico 5.1.4, porém permite que o problema principal seja selecionado, facilitando a simulação de mais de um sistema elétrico ao mesmo tempo, conforme mostrado na Figura 5.18.

Na interface principal também é possível controlar uma simulação por meio dos botões:



Figura 5.16: Interface Principal do Software

Iniciar, Pausar e Parar. Esses botões podem ser observados na Figura 5.18, esta simulação se dará sobre o momento selecionado na interface principal, porém neste tipo de controle, não é possível verificar em que estado a simulação se encontra, não é possível selecionar o algoritmo de cálculo utilizado, sendo neste caso usado o módulo de cálculo *default* do sistema e, neste tipo de controle, o software salva automaticamente a simulação no banco de dados.

Na interface principal, Figura 5.18, também estão disponíveis campos para guardar descrições do modelo elétrico carregado pelo usuário, dos momentos deste modelo e de suas simulações. Para modelo de rede elétrica é possível armazenar um nome, uma data, e uma descrição. No campo "*Momentos do Circuito*" é exibida uma lista dos momentos da rede, ao selecionar o momento, uma descrição do momento é exibida no campo "*Descrição do Momento*". Ao fazer

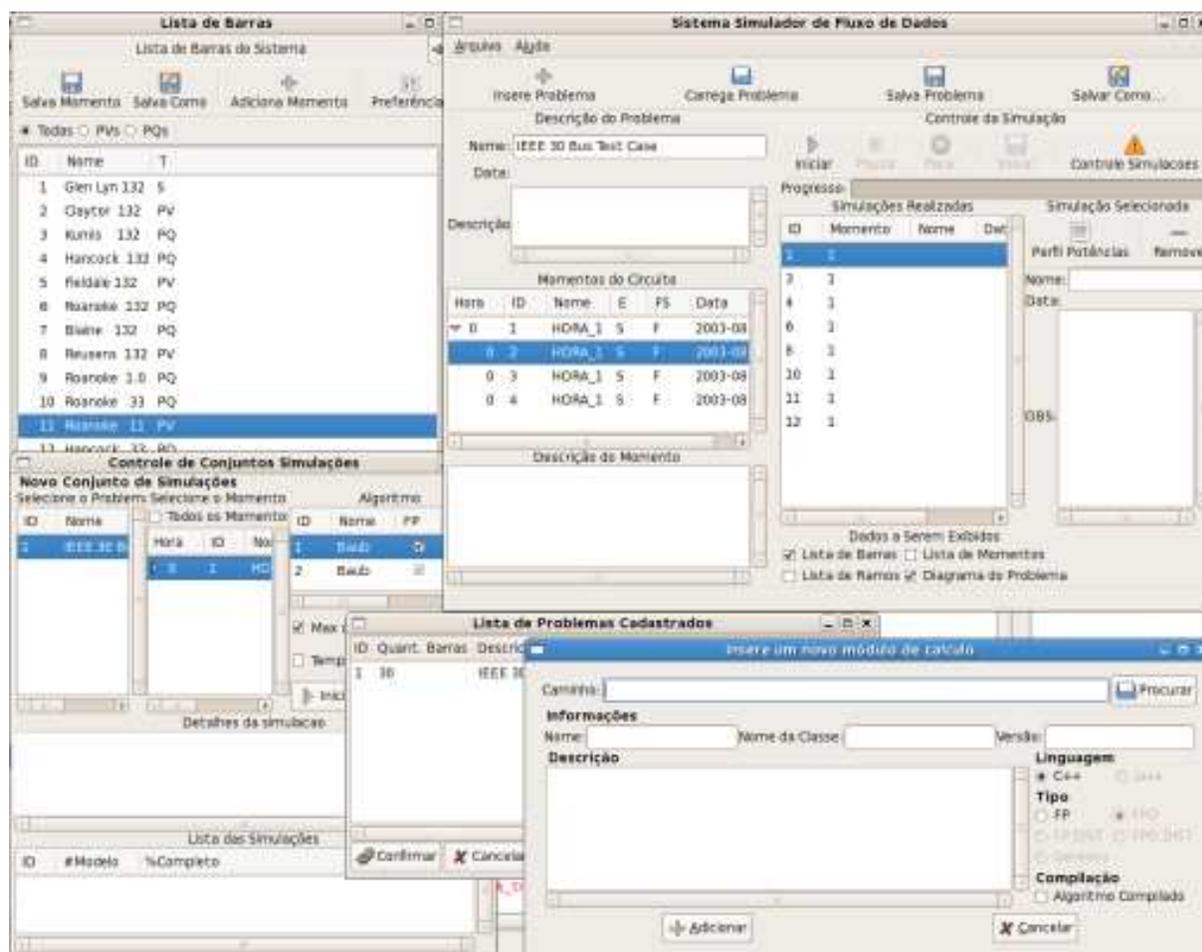


Figura 5.17: Resultado da Abordagem SDI

alguma alteração nesses campos os botões "Salva Problema" e "Salva Como ..." são ativados, permitindo que o usuário salve as alterações realizadas, ou replique o modelo de rede que esta sendo estudado, porém levando em consideração as alterações realizadas.

É disponibilizado o campo "Simulações realizadas" que exibe uma lista de todas as armazenadas no banco de dados para o modelo de rede elétrica que esta carregado no software, para cada simulação selecionada são carregados os campos "Nome", "Data" e "OBS". Ao se realizar quaisquer alterações nestes campos torna-se ativo o botão "Salvar" que permite que estas alterações sejam fixadas no banco de dados.

Ainda na Figura 5.18 é possível observar o campo "Dados a Serem Exibidos", que

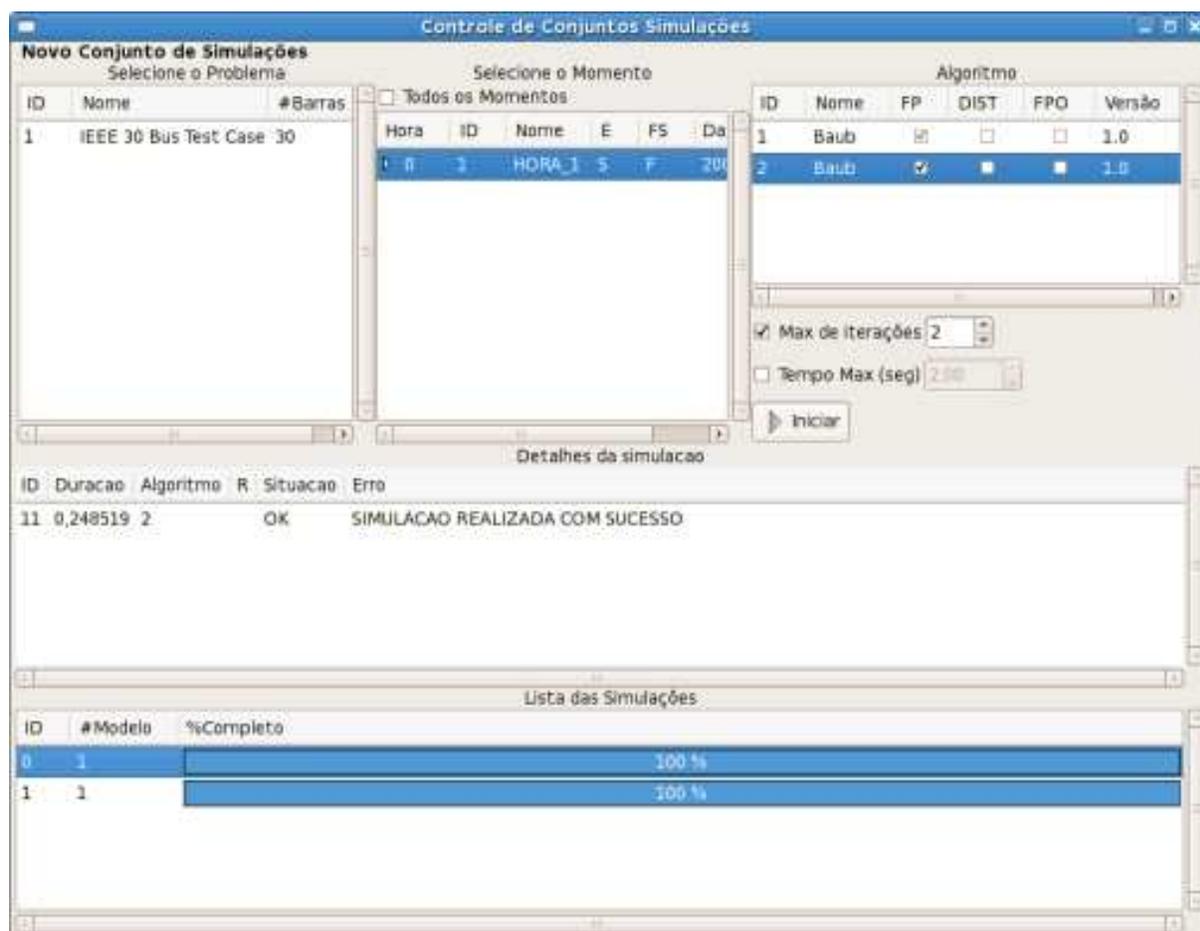


Figura 5.18: Interface Principal para Controle de Simulação

possui quatro opções: "*Lista de Barras*", "*Lista de Ramos*", "*Lista de Momentos*", "*Diagrama do Problema*". Ao se selecionar qualquer uma dessas opções uma nova janela é inicializada, quando a caixa muda de estado e deixa de estar selecionada, a janela que ela inicializou é fechada. A abertura dessas janelas ao mesmo tempo pode fazer que o usuário enfrente situações como a mostrada na Figura 5.17.

A opção "*Lista de Barras*" exibe uma janela com todas as barras do sistema elétrico, permitindo um controle detalhado, por parte do usuário, dos dados elétricos desses elementos. A caixa de seleção "*Lista de Ramos*" exibe todos os circuitos, permitindo, assim como em "*Lista de Barras*", a manipulação de todos os seus dados.

Com a opção "*Lista de Momentos*" é exibida uma interface que permite um controle

mais detalhados dos momentos do sistema, permitindo que sejam inseridos novos momentos a partir de arquivos de dados, assim como uma duplicação dos momentos já existente.

A opção "*Lista de Barras*" exibe o diagrama unifilar do modelo elétrico, assim como descrito na Seção 5.1.1, é possível navegar pela vizinhança das barras, porém neste caso cada nova vizinhança é exibida em uma nova janela, desta forma pode-se visualizar, ao mesmo tempo, a vizinhança de diversas barras.

5.3 Inserção de novos módulos de cálculo

No sistema desenvolvido está definida uma interface que permite que novos algoritmos sejam inseridos como módulos de cálculo dentro do sistema, para isto basta que ao implementar o algoritmo sejam respeitados os padrões definidos pela interface de acesso aos recursos do sistema.

Baseado no conceito de orientação a objeto, os módulos a serem inseridos devem ser implementados em linguagem C++, ou Java, onde sua classe principal, aquela que será inserida no sistema, é derivada da interface de software definida de acordo com o tipo de módulo a ser inserido. No protótipo implementado, o módulo de cálculo a ser inserido só pode ser derivado da classe *AlgoritmoBaseFP* apresentada na Seção 4.4.6.

O construtor do módulo a ser inserido deve ter como único parâmetro um ponteiro para uma instância da classe *ListaObjetos*, apresentada no tópico 4. Este ponteiro deve ser fornecido como parâmetro ao construtor da interface principal, representada pela classe *AlgoritmoBaseFP*. Não há a necessidade de que o algoritmo navegue pela estrutura da classe *ListaObjetos* para que tenha acesso aos dados do sistema.

A forma correta de obter os dados se dá através do acesso aos métodos fornecidos pela classe *AlgoritmoBaseFP*. A Tabela 5.1 relaciona os métodos de acesso aos dados do modelo elétrico fornecidos por esta classe, enquanto as Tabelas 5.2 e 5.3 relacionam os métodos disponíveis para informar os dados obtidos nas simulações. Esta classe ainda deriva uma classe mais básica, definida como *AlgoritmoBase*, conforme descrito na Seção 4.4.6.

Através dos métodos da Tabela 5.1 o módulo tem acesso as quantidades de cada tipo de elementos elétricos presentes no modelo que esta sendo simulado. Porém, de posse desses valores,

Tabela 5.1: Métodos da Classe AlgoritmoBaseFP para obtenção de informações do modelo

Método	Descrição
int getQuantRamos()	Método que retorna a quantidade de ramos presentes no modelo.
int getQuantTransf()	Método que retorna a quantidade de transformadores presentes no modelo.
int getQuantBarras()	Método que retorna a quantidade de barras presentes no modelo.
Ramo* getRamo(int)	Método que retorna a instância do ramo que teve seu índice fornecido como parâmetro, permitindo assim obter todos os dados deste determinado ramo. Os índices dos ramos variam de [0,getQuantRamos() - 1). Este método retorna <i>NULL</i> quando não for possível obter o objeto.
Transformador* getTransf(int)	Método que retorna a instância do transformador que teve seu índice fornecido como parâmetro, permitindo assim obter todos os dados deste determinado transformador. Os índices dos transformadores variam de [0,getQuantTransf()). Este método retorna <i>NULL</i> quando não for possível obter o objeto.
Barra* getBarra(int)	Método que retorna a instância da barra que teve seu índice fornecido como parâmetro, permitindo assim obter todos os dados deste determinado barra. Os índices das barras variam de [0,getQuantBarras() - 1). Este método retorna <i>NULL</i> quando não for possível obter o objeto.

é possível acessar uma instância de cada um desses objetos, fornecendo o índice do elemento como parâmetro, como pode ser observado nos métodos Ramo* getRamo(int), Transformador* getTransf(int) e Barra* getBarra(int). Ao ser fornecer um índice inválido é retornado um ponteiro nulo.

O módulo de cálculo, de posse dessas instâncias, pode acessar todos os dados elétricos dos elementos que compõem a rede de transmissão de energia.

Através dos métodos fornecidos na Tabela 5.2 o módulo de cálculo informa ao software os resultados obtidos nas simulações, fornecendo como parâmetros o índice do barra, sendo o mesmo índice usado para acessar a instância, e o valor obtido para cada tipo de dado.

Tabela 5.2: Métodos da Classe *AlgoritmoBaseFP* para salvar dados das barras obtidos nas simulações

Método	Descrição
<code>bool setTensao(int,double)</code>	Permite, ao fim da simulação, informar o valor de tensão obtido para determinada barra.
<code>bool setAnglo(int,double)</code>	Permite, ao fim da simulação, informar o valor da tensão angular obtida para determinada barra.
<code>bool setPMw(int,double)</code>	Permite, ao fim da simulação, informar o valor de potência ativa obtida para determinada barra.
<code>bool setQMVar(int,double)</code>	Permite, ao fim da simulação, informar o valor de potência reativa obtida para determinada barra.
<code>bool setTensao(int,int,double)</code>	Permite, durante a simulação, informar o valor de tensão obtido, a cada iteração, para determinada barra.
<code>bool setAnglo(int,int,double)</code>	Permite, durante a simulação, informar o valor da tensão angular obtida, a cada iteração, para determinada barra.
<code>bool setPMw(int,int,double)</code>	Permite, durante a simulação, informar o valor de potência ativa obtida, a cada iteração, para determinada barra.
<code>bool setQMVar(int,int,double)</code>	Permite, durante a simulação, informar o valor de potência reativa obtida, a cada iteração, para determinada barra.

Para informar os resultados das simulações obtidos para os circuitos usa-se os métodos definidos na Tabela 5.3, onde o primeiro parâmetro fornece o índice do circuito, o segundo parâmetro refere-se aos dados obtidos nas simulações e o terceiro parâmetro funciona como uma verificação semântica do circuito, onde o software solicita que o módulo de cálculo informe a natureza deste circuito, se trata-se uma linha de transmissão ou de um transformador.

É também possível observar nesta tabela uma replicação dos métodos, porém usando um polimorfismo onde existe mais um parâmetro, este parâmetro refere-se a iteração. Através destes métodos o módulo de cálculo pode informar os valores obtidos a cada iteração. Abordagem analoga é usada para as barras, assim, como listados na Tabela 5.2, existem métodos que permitem que sejam salvos os resultados a cada iteração.

Os métodos fornecidos por *AlgoritmoBase* podem ser observados na Tabela 5.4, assim como *AlgoritmoBaseFP* ela é uma classe abstrata, que obriga que as classes dela derivam implementem os métodos *void terminar()* e *bool run()*, como a classe *AlgoritmosBaseFP* não im-

Tabela 5.3: Métodos da Classe AlgoritmoBaseFP para salvar dados dos ramos e transformadores obtidos na simulação

Método	Descrição
bool setPkmMW(int,double,bool ramo)	Permite, ao fim da simulação, informar o valor de potência ativa de k a m obtido para determinado ramo, ou transformador, de acordo com o valor do campo ramo.
bool setPkmMW(int,double,bool ramo)	A mesma descrição do campo anterior.
bool setQkmMVAR(int,double,bool ramo)	Permite, ao fim da simulação, informar o valor de potência reativa de k a m obtido para determinado ramo, ou transformador, de acordo com o valor do campo ramo.
bool setQkmMVAR(int,double,bool ramo)	A mesma descrição do campo anterior.
bool setPkmMW(int ,int,double,bool ramo)	Permite, durante a simulação, informar o valor de potência ativa de k a m obtido, em uma determinada iteração, para determinado ramo, ou transformador, de acordo com o valor do campo ramo.
bool setPkmMW(int ,int,double,bool ramo)	A mesma descrição do campo anterior.
bool setQkmMVAR(int ,int,double,bool ramo)	Permite, durante a simulação, informar o valor de potência reativa de k a m obtido, em uma determinada iteração, para determinado ramo, ou transformador, de acordo com o valor do campo ramo.
bool setQkmMVAR(int ,int,double,bool ramo)	A mesma descrição do campo anterior.

plementa nenhum desses métodos, o módulo a ser inserido acaba obrigado a implementá-los.

A implementação destes métodos é importante porque constituem a forma principal de comunicação do sistema com os módulos. Por exemplo, ao terminar o limite de tempo que o usuário definiu para a execução da simulação o método *terminar()* é invocado, permitindo assim, que o módulo realize as operações necessárias antes de realmente efetivar o término do algoritmo.

Tabela 5.4: Métodos da Classe AlgoritmoBase

Método	Descrição
run()	Método chamado pelo sistema para informar o módulo de cálculo para iniciar a simulação
setQuantIte(int)	O módulo de cálculo informa, através deste método, a quantidade de iterações realizadas
setQuantMaxIte(int)	Usado pelo sistema para informar a quantidade máxima de iterações permitida pelo usuário
setErroTolerancia(double)	Usado pelo sistema para informar o valor de tolerância aceito pelo usuário
setErroExecucao(int)	Método que permite ao módulo implementado definir a ocorrência de um erro
double getErroTolerancia()	Forma de o módulo implementado obter o valor de tolerância
int getQuantMaxIte()	Forma de o módulo implementado obter a quantidade máxima de iterações definido pelo usuário
double getKvBase()	Maneira de o módulo implementado obter o valor Kv Base.

Neste momento é necessário que somente operações de fundamental importância sejam realizadas, pois o sistema pode forçar o algoritmo a terminar caso ele não retorne do método *terminar()*.

O número máximo de iterações definidas pelo usuário nas interfaces representadas pelas figuras 5.15 e 5.18 pode ser acessado pelo módulo de cálculo através do método *getQuantMaxIte()*.

O respeito a este limite, porém, é totalmente dependente da implementação do módulo, o sistema somente informa, mas não controla, a quantidade de iterações que estão sendo realizadas pelo algoritmo implementado no módulo.

O valor de erro que o usuário deseja para os cálculos pode ser acessado pelo método *double getErroTolerancia()*, cabendo também ao módulo de cálculo respeitar este limite.

O método mais importante, contudo, é o *run()*. É através da chamada deste método que o sistema informa ao algoritmo que os dados a ele fornecidos são válidos e que a simulação sobre estes dados deve ser iniciada. O valor de retorno do algoritmo informa ao sistema se a simulação foi, ou não, bem sucedida.

Tabela 5.5: Constantes para retorno de valores de erro

Constante	Descrição
NO_ERRO	Código usado para informar quando não houve erro durante a simulação.
ERRO_DIVERGENCIA	Código usado para informar quando o cálculo divergiu.
ERRO_VALORES	Código usado para informar quando os dados fornecidos pelo sistema não podem ser tratados pelo método de cálculo.
ERRO_LIMITE_ITE	Código usado para informar quando a rotina de cálculo não convergiu devido ao limite de iterações
ERRO_DESCONHECIDO	Código usado para informar quando um erro não citado anteriormente ocorrer.

Como pode ser observado nas Tabelas 5.1, 5.2, 5.3 e 5.4 de métodos fornecidos aos módulos fica a cargo do módulo implementado o fornecimento de dados que permitam ao usuário avaliar o status final da simulação, como por exemplo, o número de iterações realizadas, quando da ocorrência de erro, qual o tipo de erro (pelo método *setErroExecucao(int)*), para isto são definidas constantes que podem ser observadas na Tabela 5.5.

5.3.1 Interface para a inserção de novos módulos

A inserção de novos módulos ao sistema pode ser feita através da interface exibida na Figura 5.19. Esta interface é acessada através do menu *File* que pode ser observado na Figura 5.16. Deve ser fornecido o caminho para o arquivo que contém o módulo e a linguagem em que o módulo foi implementado. Além disso, pode-se inserir o nome como o módulo será referenciado nas outras interfaces do sistema, o nome da classe principal do módulo, a versão e uma pequena descrição do algoritmo.

Além destas opções há ainda a possibilidade de que o sistema receba o código fonte do módulo e realize automaticamente a compilação deste módulo. Os módulos são carregados ao sistema de acordo com o conceito de biblioteca de linkagem dinâmica.

Imediatamente após ser inserido no sistema o módulo já pode ser usado nas simulações.

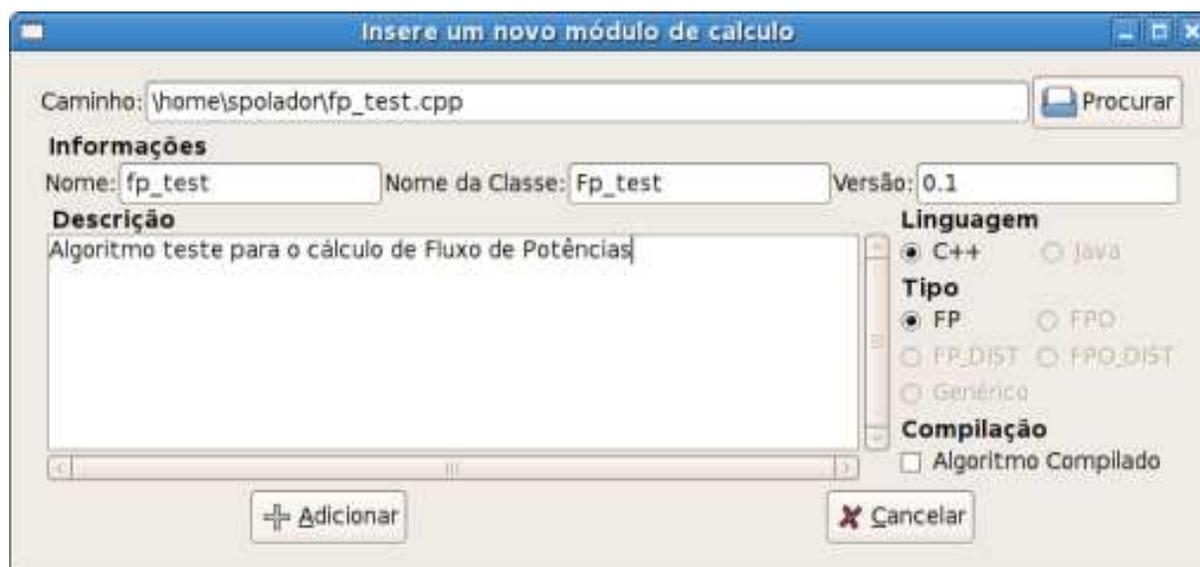


Figura 5.19: Interface para Inserção de Módulos de Cálculo

Ao ser inserido no sistema as informações fornecidas na interface da Figura 5.19 e um *checksum*², do binário, são armazenados no banco de dados, assim, mesmo que o sistema seja reiniciado não é necessário inserir o mesmo módulo novamente.

A biblioteca que constitui fisicamente o módulo deve estar na mesma pasta em que o binário principal do sistema, ou nos caminhos pré-definidos onde o sistema operacional procura, por padrão, as bibliotecas que são ligadas dinamicamente.

Qualquer alteração na biblioteca faz com que ela tenha de ser inserida novamente no sistema, pois ao inserir uma nova biblioteca o sistema faz um *checksum* sobre a biblioteca e só carrega o módulo novamente caso esse *checksum* não tenha se alterado.

5.4 Conclusão

Neste capítulo procurou-se mostrar os resultados das discussões dos capítulos 2 e 3 assim como do projeto computacional apresentado no Capítulo 4.

²Valor calculado através de uma algoritmo para se verificar a integridade do arquivo.

Foi descrito como foram implementadas as interfaces gráficas com o usuário para cada uma das funcionalidades, assim como foi demonstrado os aspectos que permitem a análise e a simulação dos dados dos sistemas de potência. Procurando desta forma mostrar como o protótipo do sistema proposto pode funcionar como uma ferramenta no auxílio de tarefas complexas de operadores e engenheiros de planejamento.

Foram apresentadas as funcionalidades que permitem comparar mais de uma simulação, assim como funcionalidades que permitem alterar dados do sistema de potência e verificar como os vários aspectos do sistema se comportam sob tais condições, permitindo que decisões sejam tomadas.

Outra funcionalidade descrita diz respeito à inserção de novos módulos de cálculo dentro do sistema. Como isso pode ser feito de uma maneira fácil e transparente.

Capítulo 6

Conclusão e trabalhos futuros

Com este trabalho procurou-se mostrar como um software desenvolvido para simulação de sistemas de transmissão/geração pode ser útil para facilitar o entendimento das simulações e dos estudos dos sistemas de potência por parte de operadores, engenheiros, pesquisadores e estudantes.

O software proposto procura fornecer recursos para análise e simulação dos sistemas de potência. Entre eles destacamos as interfaces amigáveis com o usuários, os recursos de simulação que permitem que novos módulos de cálculo sejam inseridos e o sistema de gerenciamento de dados.

Um software moderno para a área de sistemas de potência requer um conjunto de interfaces amigáveis com o usuário, pois a quantidade de dados para análise é muito grande e essas interfaces tem a função de auxiliar na realização de tarefas de alta complexidade cognitiva [KML03].

Foi desenvolvido também uma interface de software que permite a inserção de novos módulos de cálculo, com isso procuramos criar a possibilidade que estratégias de solução e algoritmos fossem testadas e comparados usando os recursos de análise desenvolvidos no software.

A modelagem e organização de uma base de dados buscaram dar a possibilidade da criação de diversos cenários de simulação, da organização desses cenários para que as comparações fossem possíveis, facilitando o processo de decisão por parte dos usuários do software.

É importante ressaltar que todos os recursos de software utilizados, tanto no projeto

quanto na implementação do protótipo são distribuídos de acordo com a licença GPL, ou uma de suas variantes, o que acabou por não gerar custo de aquisição de licenças dos produtos utilizados.

6.1 Trabalhos Futuros

Dentre as melhorias que podem ser realizadas no sistema proposto e novas funcionalidades que podem ser inseridas, podemos destacar o tratamento de dados para permitir a inserção de diversos modelos de sistemas de potência. O tratamento dos dados também permitiria uma fácil interconexão com diversos tipos de base de dados.

Dentre as ferramentas de análise desenvolvidas podem ser inseridos novos recursos, como novos gráficos que possibilitem análises de outros aspectos dos sistemas de potência. Além disso pode-se aumentar o poder de representação das redes elétricas dentro do sistema, implementando o suporte a recursos como GIS¹ o que possibilitaria a representação gráfica de maiores parcelas da rede elétrica. Para isto o software proposto foi estruturado para permitir uma fácil evolução incremental.

O software, incluindo sua base de dados, deve ser preparado para permitir que sejam considerados dados de redes de distribuição. Isso permitirá que sejam feitas análises conjuntas, possibilitando a junção das simulações os sistemas elétricos de distribuição e transmissão. As adaptações também serão necessárias no controle de inserção de novos módulos de cálculo e nas ferramentas de análise.

Outro aspecto a ser considerado diz respeito à necessidade de novas abordagens de análise que levem em consideração as necessidades das rotinas de cálculo de fluxo de potência ótimo.

A construção de um conjunto de classes especializadas de acordo com as características inerentes aos cálculos matemáticos dos sistemas de potência também pode ser implementada. Assim os módulos de cálculo a serem inseridos no sistema não teriam somente um conjunto de interfaces para se comunicar com o sistema, mas um conjunto de recursos que permitissem uma implementação mais eficiente em termos computacionais.

¹Do inglês: *Geographic Information System*

Referências Bibliográficas

- [Bau05] G. H. S. Baub. Calculo de fluxo de carga em sistemas de transmissão com alimentadores primarios de distribuição. Master's thesis, Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e Computação, 2005.
- [BPW99] T. Berry, R. Partridge, and G.S. Wang. Innovative graphical interfaces using object databases for powersystem analysis and control. In *International Conference on Human Interfaces in Control Rooms, Cockpits and Command Centres*, pages 200–205, June 1999.
- [Fow03] M. Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison Wesley, third edition edition, September 2003.
- [Gal94] R. S. Gallagher. *Computer Visualization: Graphics Techniques for Engineering and Scientific Analysis*. Solomon Press, 1994.
- [GHRV00] E. Gamma, R. Helm, J. Ralph, and J. Vlissides. *Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos*. Bookman, 2000.
- [GSJ94] J. J. Grainger and W. D. Stevenson Jr. *Power System Analysis*. McGraw-Hill, Inc., 1994.
- [Hau99] A. J. Hauser. Visualizations of global power system states in a compact and task oriented way. In *13th Power System Computation Conference*, 1999.
- [HJAA98] N. Hatziargyriou, J. Jaszczyński, G. Atsaves, and D. Agoris. Building guis for interactive network analysis. *IEEE Computer Applications in Power*, 11:46–51, October 1998.

- [IEE] Power systems test case archive, electrical engineering department, university of washington website, disponível em: <http://www.ee.washington.edu/research/pstca/>.
- [Jon05] M. T. Jones. *GNU/Linux Application Programming*. Charles River Media, Inc., 2005.
- [KML03] F. Kober, A. Manzoni, and F. A. B. Lemos. An objected-oriented approach to development and integration of graphical user interface and power system framework. In *IEEE Bologna PowerTech Conference*, 2003.
- [Kum05] R. R. Kumar. *Human Computer Interaction*. Firewall Media, 2005.
- [Lar01] C. Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*. Prentice Hall, second edition, 2001.
- [Lav00] M. Lavergne. Graphical user interface for next generation power systems. In *Telecommunications Energy Conference, 2000. INTELEC. Twenty-second Internacional*, pages 109–112, 2000.
- [LJC⁺88] X. Li, J. Jiang, J. R. Cantwell, J. R. Bourne, and K. Kawamura. Graphics-based qualitative simulation generator for power distribution systems. In *Proceedings of the 1st international conference on Industrial and engineering applications of artificial intelligence and expert systems*, volume 2, pages 877–884, 1988.
- [LO97] M. Loukides and A. Oram. *Programming with GNU Software*. O’Reilly & Associates, Inc., first edition, January 1997.
- [LQ98] Y. Liu and J. Qiu. Visualizations of power system static security assessment based on gis. In *Internacional Conference of Power System Technology*, volume 2, pages 1266–1270, Aug 1998.
- [MDS01] D. R. Musser, G. J. Derge, and A. Saini. *C++ Programming with the Standard Template Library*. Addison-Wesley, second edition, 2001.
- [Mon83] A. Monticelli. *Fluxo de Carga em Redes de Energia Elétrica*. Edgard Blucher, 1983.
- [MS03] N. M. Maricar and S. Shalidehpour. An engineering analysis tool in power-system using object-oriented graphical integrator. in *Proc. National Power and Energy Conference of Malaysia*, pages 41–51, 2003.

- [Mue02] J. P. Mueller. *Visual C++ .NET Developers Guide*. McGraw-Hill, Inc., 2002.
- [MWKC06] B. A. Myers, D. A. Weitzman, A. J. Ko, and D. H. Chau. Answering why and why not questions in user interfaces. In *ACM Conference on Human Factors in Computing Systems*, 2006.
- [Ove03] T. J. Overbye. Fostering intuitive minds for power system desing. *IEEE power m& energy magazine*, pages 38–43, 2003.
- [Pen99] H. Pennington. *GTK+ / Gnome Application Development*. New Riders Publishing, first edition, 1999.
- [Pre95] R. S. Pressman. *Engenharia de Software*. Makron Books, 1995.
- [RB00] H. V. Rocha and M. C. C. Baranauskas. *Design e avaliação de interfaces humano-computador*. IMW-USP, 2000.
- [RJB04] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison Wesley, second edition, 2004.
- [Saa02] H. Saadat. *Power System Analysis*. McGraw-Hill, Inc., 2002.
- [Sar02] A. Saraiva. Papel dos controles secundários numa análise de estabilidade de tensão em regime permanente. Master's thesis, Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e Computação, 2002.
- [Sch98] H. Schildt. *C++: Complete Reference*. McGraw-Hill, Inc., third edition, 1998.
- [SG93] L. S. Steinberg and D. H. Gitomer. Cognitive task analysis, interface design, and technical troubleshooting. In *Proceedings of the 1st international conference on Intelligent user interfaces*, pages 185–191, 1993.
- [SJ82] W. D. Stevenson Jr. *Elements of Power System analysis*. McGraw-Hill, Inc., fourth edition, 1982.
- [SLI99] J. Shin, W. Lee, and D. Im. A windows-based interactive and graphic package for the educationand training of power system analysis and operation. *IEEE Transactions on Power System*, 14(4):1193–1199, November 1999.

- [SO04] Y. Sun and T. J. Overbye. Visualizations of power system contingency analysis data. *IEEE Transactions on Power Systems*, 19(4):1859–1866, november 2004.
- [Sto74] B. Stott. Review of load-flow calculation methods. *Proceedings of the IEEE*, 62:916–929, July 1974.
- [Str97] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley, third edition, 1997.
- [TH57] W. F. Tinney and C. E. Hart. Power flow solution by newton method. *IEEE Transaction on Power Apparatus and Systems*, PAS-86:1449–1456, 1957.
- [TKM⁺00] Y. Tada, A. Kurita, M. Masuko, Y. Takahar, and K. Koyanagi. Development of an integrated power system analysis package. In *International IEEE Conference on Power System Technology*, volume 3, pages 1695–1700, 2000.
- [weba] Easypower powerflow, esa company, disponível em: <http://easypower.com/>.
- [webb] Interactive simulation package for electrical networks (ispen), eps software nach mass, disponível em: <http://www.eps.ch/50179596ad0c68204/index.html>.
- [webc] Power systems analysis framework (psaf), powersys, disponível em: <http://www.powersys.fr/psafpresent.php>.
- [webd] Powerfactory optimal power flow (opf), digsilent gmbh, disponível em: <http://www.digsilent.de/software/>.
- [Webe] Powerworld simulator, powerworld corporation website, disponível em: <http://www.powerworld.com/>.
- [XLKL99] W. Xu, Y. Liu, D. Koval, and M. A. Lipset. Using spreadsheet software as a platform for power system analysis. *IEEE Computer Applications in Power*, 12:41–45, January 1999.