

DIEGO DE CASTRO GRILLO

**Desenvolvimento e Estudo de
usabilidade de uma Interface Gráfica
para Visualização e Manipulação de
Dados Oscilográficos no Padrão IEEE
COMTRADE**



São Paulo

2012

Diego de Castro Grillo

***DESENVOLVIMENTO E ESTUDO DE
USABILIDADE DE UMA INTERFACE
GRÁFICA PARA VISUALIZAÇÃO E
MANIPULAÇÃO DE DADOS
OSCILOGRÁFICOS NO PADRÃO IEEE
COMTRADE***

Trabalho de Conclusão de Curso apresentado à Escola de Engenharia de São Carlos, da Universidade de São Paulo. Curso de Engenharia da Computação com ênfase em Sistemas Computacionais Avançados.

Orientador:

Prof. Dr. Mário Oleskovicz

São Carlos
2012

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

G85955 9d Grillo, Diego de Castro
Desenvolvimento e Estudo de usabilidade de uma
Interface Gráfica para Visualização e Manipulação de
Dados Oscilográficos no Padrão IEEE COMTRADE / Diego de
Castro Grillo; orientador Mário Oleskovicz. São Carlos,
2012.

Monografia (Graduação em Engenharia de Computação)
-- Escola de Engenharia de São Carlos da Universidade
de São Paulo, 2012.

1. Percurso Cognitivo. 2. Interface Gráfica. 3.
Padrão COMTRADE. 4. Oscilografias. 5. Qualidade da
Energia Elétrica. 6. Proteção Digital de Sistemas
Elétricos de Potência. I. Título.

FOLHA DE APROVAÇÃO

Nome: Diego de Castro Grillo

Título: “Desenvolvimento e estudo de usabilidade de uma interface gráfica para visualização e manipulação de dados oscilográficos no padrão IEEE COMTRADE”

Trabalho de Conclusão de Curso defendido em 29 / 11 / 2012.

Comissão Julgadora:

Resultado:

Profa. Dra. Sarita Mazzini Bruschi
SSC/ICMC/USP

APROVADO

Dr. Ulisses Chemin Netto
SEL/EESC/USP

Aprova L

Orientador:

Prof. Dr. Mário Oleskovicz - SEL/EESC/USP

Coordenador pela EESC/USP do Curso de Engenharia de Computação:

Prof. Associado Evandro Luís Linhari Rodrigues

*Este trabalho é dedicado à minha família,
cujo amor e sabedoria me inspiram
a melhorar dia após dia.*

Agradecimentos

Ao término deste trabalho, deixo aqui meus sinceros agradecimentos:

- a Deus por tudo;
- ao Prof. Dr. Mário Oleskovicz, por toda dedicação, paciência e estímulo em sua orientação;
- a todos os professores da Escola de Engenharia de São Carlos - EESC, da Universidade de São Paulo - USP;
- a minha família, pelo incentivo e segurança que me passaram durante todo esse período;
- aos amigos do curso de Engenharia de Computação pelo agradável convívio; e
- a todos que direta ou indiretamente contribuíram para a realização deste trabalho.

*“O único lugar onde o sucesso vem antes
do trabalho é no dicionário.”*

Albert Einstein

Resumo

GRILLO, D. C. Desenvolvimento e Estudo de usabilidade de uma Interface Gráfica para Visualização e Manipulação de Dados Oscilográficos no Padrão IEEE COMTRADE, 2012. p.70. Trabalho de Conclusão de Curso (Engenharia de Computação com Ênfase em Sistemas Computacionais Avançados) - Escola de Engenharia de São Carlos (EESC), Universidade de São Paulo, São Carlos - SP, 2012.

Este trabalho apresenta a utilização em *software* do método de análise de interfaces gráficas denominado “Percurso Cognitivo”, aplicado para visualização e manipulação de dados oscilográficos armazenados no padrão IEEE COMTRADE. O padrão IEEE COMTRADE é comumente utilizado por uma diversidade de equipamentos, relacionados, por exemplo, às áreas de qualidade da energia elétrica e proteção digital de sistemas elétricos. Desta diversidade de aplicações, tem-se então a necessidade da criação de *softwares* que atuem sobre esse padrão, com o objetivo de facilitar e enriquecer os estudos e as análises do desempenho e da operação dos sistemas elétricos de potência. Neste sentido, é de fundamental importância que o aplicativo atuante apresente uma boa usabilidade, para ser aceito no ambiente em que será inserido. Por isso é desejável uma interface gráfica agradável e facilitadora em relação à sua aprendizagem. O percurso cognitivo é uma técnica de avaliação de interfaces que apresenta atenção especial para o suporte dado na aprendizagem exploratória do *software* pelo usuário. Como resultado da aplicação do método, tem-se uma série de melhorias que podem ser incorporadas à interface gráfica de um *software*. Este documento descreve a aplicação desse método e o resultado da implementação dessas melhorias buscando mostrar ao leitor o quanto a utilização do percurso cognitivo pode enriquecer a interface gráfica de uma dada aplicação. Como resultado, tem-se uma interface gráfica de maior usabilidade com uma avaliação positiva frente ao método aplicado, destacando sua eficiência e eficácia.

Palavras Chave: Percurso Cognitivo, Interface Gráfica, Padrão COMTRADE, Oscilografias, Qualidade da Energia Elétrica, Proteção Digital de Sistemas Elétricos de Potência.

Abstract

GRILLO, D. C. Development and Usability Study of a Graphical Interface for Viewing and Manipulating Oscillographic Data in the IEEE COMTRADE Standard, 2012. p.70. Trabalho de Conclusão de Curso (Engenharia de Computação com ênfase em Sistemas Computacionais Avançados) - Escola de Engenharia de São Carlos (EESC), Universidade de São Paulo, São Carlos - SP, 2012.

This work presents the software usage of the graphical user interface analysis method called “Cognitive Walkthrough”, used for viewing and manipulating oscillographic data in the IEEE COMTRADE standard. The IEEE COMTRADE standard is usually employed by a range of devices, related, for example, to the fields of electric power quality and protection of digital electric systems. From this diversity of applications, there is the need to create software that operate on this standard, in order to enhance and make easier the studies and analysis of the performance and operation of electric power systems. In this sense, it is essential for the active application to feature a good usability, in order to be accepted in the environment in which it will be inserted. Therefore it is desirable a pleasant user-friendly graphical user interface. The Cognitive Walkthrough is a graphical user interface evaluating technique that focus on the support given to the software’s exploratory learning by the user. As a result of applying this method, there is a variety of improvements which can be incorporated into a software’s graphical user interface. This paper describes the application of this method and the results of implementing these improvements, attempting to show the reader how much the usage of the Cognitive Walkthrough can enhance the graphical interface of a given application. As a result it has a graphical interface for greater usability with a positive rating referring to the method applied, highlighting its efficiency and effectiveness.

Keywords: Cognitive Walkthrough, Graphical Interface, COMTRADE Standard, Oscillographic Records, Power Quality, Digital Protection of Power Systems.

Sumário

Resumo	ix
Abstract	xi
Lista de Abreviaturas e Siglas	xvi
Lista de Figuras	xviii
1 Introdução	1
1.1 Contexto	1
1.2 Definição do problema	3
1.3 Objetivos do trabalho	6
1.4 Organização do trabalho	7
2 Revisão Bibliográfica	9
2.1 Desenvolvimento de uma interface flexível e amigável	9
2.2 <i>Cognitive Walkthrough</i> (Percurso Cognitivo)	11
2.3 O padrão COMTRADE	13
2.4 Aplicativos Existentes	14
3 A Ferramenta WAPS	21
3.1 Descrição do Módulo 1	21
3.2 Descrição do Módulo 2	28
4 Metodologia de Trabalho	35
4.1 Preparação do Percurso Cognitivo	35

4.2	Metodologia Incremental	38
4.3	Ferramentas computacionais aplicadas	40
5	Desenvolvimento e Resultados	43
5.1	Percurso cognitivo e definição das melhorias	43
5.2	Implementação das melhorias na interface do aplicativo e avaliação dos resultados	53
5.2.1	Tela Principal	54
5.2.2	Janela de Seleção de Canais	57
5.2.3	Janela de configuração para exibição dos gráficos	58
5.2.4	Janela de exibição de gráficos	59
5.2.5	Janela <i>Math Builder</i>	62
5.2.6	Janela de exibição de gráficos fasoriais	64
5.2.7	Janela de descarte de canais	65
6	Considerações Finais	66
	Referências	69
	Anexo	71
	Modelo de Orientação a Objetos Aplicado ao Projeto e Desenvol- vimento da Ferramenta Computacional WAPS para Mani- pulação e Visualização de Arquivos Oscilográficos	71
	Ferramenta Multi Arquivos para Visualização e Manipulação Os- cilográfica de Sinais Elétricos	81

Lista de Abreviaturas e Siglas

API	<i>Java Application Programming Interface</i>
QEE	Qualidade da Energia Elétrica
SEP	Sistemas Elétricos de Potência
WAPS	<i>Wave Analyser for Power Systems</i>
RMS	<i>Root Mean Square</i>
FFT	<i>Fast Fourier Transform</i>
FSF	<i>Free software Foundation</i>
COMTRADE	<i>IEEE Standard Common Format For Transient Data Exchange For Power System</i>
MDI	<i>Multiple Document Interface</i>
IHC	Interação Humano-Computador
PQDIF	<i>Power Quality Data Interchange Format</i>
ATP	<i>Alternative Transients Program</i>
TOP	<i>The Output Processor</i>
IDFT	<i>Inverse Discrete Fourier Transform</i>
SINAPE	Sistema Integrado de Apoio à Análise de Perturbações
JVM	<i>Java Virtual Machine</i>
GUI	Interface Gráfica do Usuário
LGPL	<i>Lesser General Public License</i>
MEC	Ministério da Educação

Lista de Figuras

1	Diagrama do WAPS	6
2	Sistema aplicado a uma usina	15
3	Interface gráfica do TOP	16
4	Interface gráfica do SIGRA	17
5	Interface gráfica do <i>.NET Relayer</i>	18
6	Interface gráfica do SINAPE	19
7	Tipos de Canais do WAPS	22
8	Seletor de arquivos	23
9	Janela de Seleção de Canais	24
10	<i>Math Builder</i>	25
11	Janela de Configuração para a Exibição de Gráficos	27
12	Janela de Descarte	28
13	Janela de exibição de gráficos	30
14	Painel de opções	31
15	Janela gráficos fasoriais	34
16	Tarefas Analisadas	37
17	Modelo Incremental	39
18	Caixa de combinação	44
19	Painel Tabulado	47
20	Janela de Fasores com Painel de Opções habilitado	50
21	Tabela seleção segundo argumento e botão de opções	52
22	Painel de Parâmetros da janela <i>Math Builder</i>	53
23	Tela de Abrir após melhoria	54

24	Árvore de Canais Carregados	55
25	Menu <i>Pop-up</i>	56
26	Melhoria Configuração para exibição de gráficos	59
27	Melhorias Janela de Exibição de Gráficos	61
28	Caixa de Dialogo <i>Math Builder</i>	63
29	Janela de Exibição de Fasores após melhorias	65

1 *Introdução*

Neste capítulo é introduzido o contexto e os conceitos iniciais que levaram ao desenvolvimento da ferramenta WAPS - *Wave Analyser for Power System*[1], além de mostrar porque é importante se preocupar com a criação de uma interface gráfica de qualidade para que o *software* em desenvolvimento seja bem aceito pelos usuários finais. Após uma breve introdução, tem-se uma seção que apresentará a definição do problema e, por fim, outra dedicada a explicar a divisão dos capítulos deste documento.

1.1 Contexto

A energia elétrica está diretamente ligada com a qualidade de vida das pessoas, com a produtividade de uma indústria e com a segurança dos aparelhos e equipamentos elétricos. Visto sua importância, é natural que um processo de discussão sobre a qualidade desse produto se inicie. É esperado que esse serviço seja contínuo, que não haja oscilações de tensão, interferência nos sistemas de comunicação, desequilíbrios, entre outros fatores que atrapalhem a sua utilização [2].

Porém as linhas de transmissão de energia elétrica estão sujeitas a falhas, e tais falhas podem causar danos em aparelhos elétricos de potência ou até acidentes.

É de importância fundamental, por exemplo, que logo após uma falha em uma linha de transmissão de energia elétrica, as medidas adequadas possam ser tomadas a fim de evitar com isso, estragar aparelhos elétricos de potência ou causar danos no sistema como um todo. Para que essas medidas possam ser tomadas, é necessário descobrir suas causas, e isso só é possível quando as informações referentes à linha de transmissão em falta são registradas e tecnicamente analisadas.

Visto a importância do estudo e entendimento dos fenômenos relacionados a falta de Qualidade da Energia Elétrica (QEE), como também da proteção de Sistemas Elétricos de Potência (SEP), diversas ferramentas computacionais para a simulação e representação de redes de energia elétrica foram desenvolvidas, buscando contribuir com o cenário descrito. Pode-se citar, por exemplo, fontes de dados de transitórios (associadas a *hardware* e a *software*), como relés digitais de proteção, programas de simulação de transitórios, simuladores analógicos entre outros, para auxiliar o corpo técnico especializado na busca por soluções de seus problemas.

Da prática, tem-se que muitas vezes os registros digitais gerados pelas ferramentas citadas seguem um formato proprietário sem nenhum tipo de padronização, o que dificulta a troca de informações entre equipamentos de fabricantes diferentes. Com isso, uma necessidade detectada ao longo do tempo foi à criação de um formato comum para arquivos de dados digitais e mídias, possibilitando o intercâmbio entre as informações referentes às simulações de sinais e também aos dados de perturbações gerados por essas ferramentas.

Todos esses esforços foram de fundamental importância para aumentar o entendimento dos fenômenos relacionados à proteção. Porém, ainda preocupado com esse ambiente, este documento destaca a importância dos *softwares* criados a fim de trabalhar em conjunto com as ferramentas computacionais descritas acima. Tais *softwares* atuam nesse contexto permitindo a visualização dos dados obtidos nas perturbações e transitórios através de gráficos, além da possibilidade de utilizar operações e ferramentas matemáticas com os mesmos. Esses aplicativos têm grande utilidade no estudo do SEP, principalmente na área de localização de faltas. Em muitos trabalhos correlatos que serão citados na revisão bibliográfica deste documento, os esforços se concentram em criar uma interface gráfica que facilite a utilização pelo usuário final e, ao mesmo tempo, proporcione liberdade para que o mesmo visualize os dados conforme sua preferência.

Porém a variedade de *softwares* atuantes nesse contexto não é muito grande, o que obriga os consumidores industriais e comerciais a pagar altos valores em licenças para assim utilizar esses produtos. Outro ponto importante detectado se refere à usabilidade e a interface desses aplicativos. Portanto, no documento que segue, serão enfatizados alguns tópicos relacionados ao desenvolvimento de interfaces de *softwares*.

Vale lembrar que um programa que não apresenta uma boa usabilidade não é utilizado. Portanto, é preciso ter a preocupação com diversos fatores na implementação de um aplicativo, tais como a funcionalidade, a velocidade, a eficiência, a confiabilidade, a segurança, a consistência dentre outros, pois o usuário facilmente percebe a falta de um desses no sistema, e essa percepção determina a não utilização do programa[3].

Alguns cuidados devem ser tomados pelos projetistas para não desagradar as pessoas que utilizam o *software*, ou seja, atender os fatores citados acima. Precisa-se saber previamente o local onde o sistema será utilizado (*hardware* envolvido), ter conhecimentos específicos sobre o assunto, conhecer o tipo de usuário, e, principalmente, não cometer erros como, por exemplo, assumir que todos os usuários são iguais ou que os usuários são como o projetista (importante lembrar que o usuário não conhece nenhum detalhe da implementação).

Como fato, tem-se que as características de fenômenos relacionados à QEE e a proteção de SEP não são facilmente visualizadas devido à quantidade de dados gerados e da magnitude de seus valores. Consequentemente, a interface gráfica com o usuário deve ser uma das prioridades no desenvolvimento de sistemas para o estudo e análise do campo em questão.

1.2 Definição do problema

As análises de fenômenos ligados à geração e transmissão de energia elétrica são de fundamental importância para garantir sua qualidade. Portanto, existe a necessidade da criação de uma ferramenta que seja capaz de tornar os dados referentes às diferentes grandezas envolvidas nesse ambiente, manipuláveis computacionalmente. Para tanto, diversos padrões foram criados para registrar as oscilografias geradas sobre os SEP, simulações, ensaios e outras perturbações. Considerando-se esta última observação, deseja-se uma ferramenta com a capacidade de interpretar os dados independentemente do padrão utilizado, pois dessa forma a escalabilidade do aplicativo, quando se trata do formato dos dados importados, será garantida. Ou seja, caso um novo padrão seja inserido no mercado, a ferramenta em questão terá a capacidade de adaptar-se ao mesmo não ficando obsoleto. É importante também que esse aplicativo tenha a capacidade de converter outros padrões já existentes no padrão inserido no mercado e vice-versa,

pois desta forma se aumenta a usabilidade do produto e, por consequência, um número maior de usuários.

Outro ponto importante é que o aplicativo não poderá ficar preso ao estudo do registro oscilográfico de um canal de uma única estação. Para isso deverá ser projetado de tal forma que haja a possibilidade de manipular vários canais de diferentes estações.

Também devem ser consideradas questões para o armazenamento. Para armazenar os registros oscilográficos de uma maneira conveniente, os sinais referentes as tensões e as correntes de um determinado canal devem apresentar uma frequência de amostragem adequada. Visto isso, é válido que o *software* exiba esses dados amostrados de uma maneira clara. Uma lista com todas as amostras relatadas não é atraente para o usuário, porém a utilização de ferramentas como gráficos e tabelas pode ser uma boa alternativa garantindo satisfação ao usuário.

Além disso, o aplicativo deverá ter a funcionalidade de realizar algumas operações algébricas julgadas importantes, por exemplo, no ambiente de proteção. Aponta-se as quatro operações básicas (adição, subtração, divisão e multiplicação) entre os diferentes pontos amostrados, a realização da Transformada Rápida de Fourier (do inglês *Fast Fourier Transform*, ou FFT) nos sinais importados para a obtenção dos espectros de frequência, a anti-transformada discreta de Fourier (IDFT - *Inverse Discrete Fourier Transform*), o valor quadrático médio (do inglês *Root Mean Square*, ou RMS) e também o fasor de cada ponto amostrado.

Outras duas características devem ser incorporadas ao *software* para que o mesmo tenha uma boa aceitação no ambiente descrito. É importante que ele seja funcional em diversas plataformas e seja livre.

De acordo com a definição da Fundação para o *Software* Livre[4] (FSF, *Free Software Foundation*):

[...] *software* livre é uma questão de liberdade dos usuários de executar, copiar, distribuir, estudar, mudar e melhorar o *software*. Mais precisamente, significa que os usuários do programa têm as quatro liberdades essenciais: * A liberdade de executar o programa, para qualquer propósito (liberdade 0). * A liberdade de estudar como o programa funciona e alterá-lo para ele fazer o que quiser (liberdade 1). Acesso ao código-fonte é uma condição para isso. * A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade 2).

* A liberdade de distribuir cópias de suas versões modificadas para os outros (liberdade 3). Ao fazer isso você pode dar a toda a comunidade a oportunidade de beneficiar de suas alterações. Acesso ao código-fonte é uma condição para isso.

O aplicativo deverá ser modular, ou seja, deverá permitir a adição de novas rotinas ou a modificação das já existentes. Note que isso só é possível quando o produto é livre conforme comentado acima.

Neste contexto, para atingir as características citadas acima, foi desenvolvido um aplicativo de *software* que está dividido em duas partes, conforme mostra a Figura 1[1].

Módulo 1: Essa parte é responsável por importar um arquivo oscilográfico escolhido pelo usuário para a conversão dos dados em um formato comum ao *software*. Esse formato comum deverá conter as mínimas informações necessárias para a realização das quatro operações básicas em cada instante de tempo da amostra, além de valores necessários para o cálculo da Transformada Rápida de Fourier, Anti-transformada de Fourier, dos valores RMS e dos fasores do sinal;

Módulo 2: Responsável pela exibição dos sinais importados e criados através das manipulações algébricas realizadas no módulo 1. Vale frisar a importância deste módulo por permitir ao usuário a visualização de maneira clara dos valores das amostras, dos fasores, dos valores RMS e dos decorrentes da aplicação da FFT em cada instante de tempo, permitindo ao mesmo, inclusive, realizar comparações e com isso caracterizar uma poderosa ferramenta para a análise dos fenômenos ligados a geração e distribuição de energia elétrica.

O trabalho de conclusão de curso desenvolvido se refere aos módulos 1 e 2 do aplicativo de *software* caracterizado acima, cujo objetivo é formar o aplicativo Wave Analyser for Power Systems (WAPS).

Uma análise para a construção da interface gráfica desses módulos será feita a partir do método de inspeção de usabilidade denominado *Cognitive Walkthrough* [5], que será explicado posteriormente nesta monografia. Esse método será empregado visando melhorar a qualidade da interface gráfica aumentando a aceitação do *software* pelos usuários.

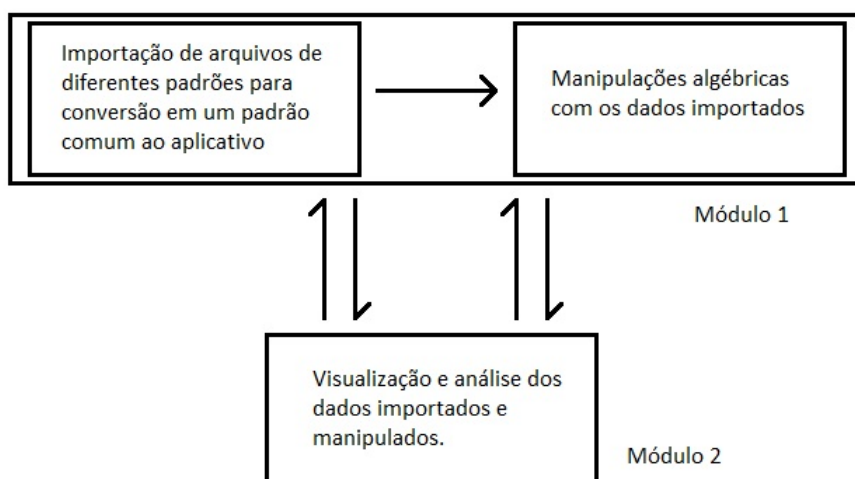


Figura 1: Diagrama do aplicativo de *software* proposto.

Segundo Nielsen[6], alguns métodos de usabilidade podem ser aplicados de forma rápida e sem gastar muitos recursos, produzindo resultados bastante satisfatórios. Mas ele também alerta que esses métodos não devem ser empregados nas fases iniciais do ciclo de vida do *software*, pois é difícil inspecionar algo que ainda não existe. Por outro lado o autor alerta também que estes métodos são pouco adequados para as fases finais do projeto, quando o projeto já foi lançado e existe apenas a pretensão em obter ideias para versões futuras. Nesse caso, são mais indicados estudos de campo, registros de operações e análise das chamadas de suporte aos usuários [7].

Visando melhorar a qualidade da interface gráfica das ferramentas de *software* de análise e visualização de registros oscilográficos, pode-se aplicar um desses métodos na mesma, buscando torná-la mais amigável àqueles que necessitam das funcionalidades oferecidas por tais ferramentas.

1.3 Objetivos do trabalho

Este trabalho tem como objetivo o estudo e desenvolvimento de uma interface gráfica de um aplicativo de *software* denominado WAPS, para a manipulação e a análise de perturbações e transitórios de redes de energia elétrica. Esta interface será desenvolvida a partir dos módulos 1 e 2 já existentes, citados na seção 1.2,

com o auxílio do método de inspeção de interfaces gráficas denominada percurso cognitivo.

1.4 Organização do trabalho

O capítulo 2 apresenta a revisão bibliográfica feita para esta monografia, que está dividida em três partes.

Desenvolvimento de uma interface flexível e amigável: Nessa seção foram levantados alguns conceitos sobre a avaliação de interfaces gráficas, bem como questões referentes à interação usuário e computador;

***Cognitive Walkthrough* (Percurso cognitivo):** Nessa seção temos algumas explicações segundo diferentes autores sobre as etapas do método de avaliação de interfaces denominado percurso cognitivo;

O Padrão COMTRADE: Serão abordados nesse tópico algumas características desse padrão utilizado pela ferramenta WAPS.

Aplicativos existentes: Para finalizar a revisão bibliográfica, essa seção mostra alguns aplicativos existentes na área de proteção de redes de energia elétrica permitindo caracterizar os *softwares* que o profissional dessa área costuma utilizar, tendo foco nos diferentes tipos de interface gráfica que tais aplicativos possuem.

O capítulo 3 descreve o funcionamento dos módulos 1 e 2 do WAPS antes da aplicação do método do percurso cognitivo. Julga-se importante entender o funcionamento desse trabalho para que o leitor consiga acompanhar toda a análise necessária durante a aplicação do método proposto.

O Capítulo 4 descreve toda a metodologia de trabalho utilizada nesse documento. A mesma está dividida em três tópicos que serão apresentados a seguir:

Preparação do Percurso Cognitivo: Aqui será levantado e definido todos os pré-requisitos necessários antes da aplicação do método de avaliação propriamente dito, considerando as características do *software* em questão (WAPS), bem como suas funcionalidades e tarefas;

Metodologia de Desenvolvimento Incremental: Idealiza-se nesse trabalho que sejam aplicadas melhorias em todas as falhas de interface gráfica encontradas através do método de avaliação utilizado. Para tanto, será utilizada a metodologia de desenvolvimento incremental na implementação das melhorias propostas. Tal metodologia será apresentada neste item;

Ferramentas aplicadas: Esta seção descreve as ferramentas utilizadas ao longo do trabalho.

O capítulo 5 é responsável por conter o desenvolvimento do trabalho proposto, juntamente com a apresentação dos resultados obtidos durante tal desenvolvimento. O mesmo foi organizado nos seguintes itens:

Percurso Cognitivo e definição das melhorias: Este item descreve como foi aplicado o método na interface gráfica WAPS, e mostra os resultados obtidos após a aplicação do mesmo. Após aplicar o método e detectar os problemas de interface, é possível definir as melhorias que deverão ser feitas na interface gráfica do *software*. Estas melhorias serão definidas nesta seção;

Implementação e avaliação das melhorias na interface do aplicativo: Este item apresentará os resultados obtidos com as melhorias implementadas na interface gráfica, em conjunto com uma breve comparação do *software* antes e depois das alterações buscando verificar o quanto é válido utilizar o método de avaliação Percurso Cognitivo.

O Capítulo 6 apresenta a conclusão do trabalho, levantando algumas dificuldades, propondo algumas ações para trabalhos futuros e, principalmente, avaliando o trabalho realizado.

2 *Revisão Bibliográfica*

Para a realização do trabalho proposto serão considerados estudos referentes ao desenvolvimento de *softwares*, Interação Humano-Computador (IHC) e aplicativos já existentes para análise das oscilografias.

2.1 Desenvolvimento de uma interface flexível e amigável

Um bom projeto de *software* é fundamental para garantir que ele realize suas funções principais, e ao mesmo tempo garanta a satisfação do usuário. Carvalho e Daltrini[8] afirmam que projetistas se preocupam muito em desenvolver um produto com excelente qualidade técnica, esquecendo-se muitas vezes que o usuário não percebe essa característica enquanto utiliza um aplicativo. O usuário avalia a qualidade do produto principalmente através da interface com o sistema. Portanto, não adianta desenvolver um produto que atenda às mais avançadas técnicas de engenharia de *software* sendo que este depois de instalado acaba sendo rejeitado pelo usuário devido à dificuldade na utilização.

Marcus[9] afirma em seu trabalho que existem três princípios fundamentais para o projeto de uma interface:

Organização: Uma interface organizada deve ser consistente, ou seja, seguir os mesmos padrões em todas as janelas, ter um bom *layout* e uma boa navegabilidade, relacionando tudo que for semelhante na tela.

Economia: Baseia-se na ideia de maximizar os resultados obtendo o maior número de ações utilizando o menor número de recursos possível.

Comunicação: Cada interface tem uma cor e textura apropriada, e precisa ser legível, dotada de símbolos; ter uma tipografia padronizada e apresentar

ao usuário visões múltiplas, ou seja, permitir que ele escolha a maneira de visualizar o conteúdo de uma tela.

Considerando a importância da interface com o usuário um novo conceito foi criado conhecido como a IHC já citada nesse documento. O MEC (Ministério da Educação)[10] define IHC como “a disciplina relacionada ao projeto, implementação e avaliação de sistemas computacionais interativos para uso humano, juntamente com os fenômenos relacionados a esse uso”.

Shneiderman [11] descreve em sua obra a importância do desenvolvimento de boas interfaces gráficas. Ele justifica tal fato através de pesquisas que mostram que um projeto de interface homem-computador proporciona melhoria no tempo de aprendizado, na taxa de erros, na velocidade de execução e na satisfação do usuário. Além disso, ele alega que projetistas reconhecem que sistemas considerados fáceis de usar geram uma margem competitiva na recuperação da informação, automação de escritório e computação pessoal.

Rocha e Baranauskas [3] definem o conceito IHC e a importância de uma boa interface. No livro, tópicos como os aspectos mentais humanos, engenharia cognitiva, ergonomia, a maneira de pensar, agir e imaginar das pessoas são discutidos para assim, entender o homem e a partir disso desenvolver uma interface que o satisfaça.

Mas não basta entender apenas o usuário, pois também é preciso ser levantado um estudo sobre o computador/*software*. Pressman[12] detalha tópicos importantes como alocação de funções, consistência, compatibilidade entre o estímulo e a resposta, expectativas, estereótipos, facilidade de aprendizagem, uso e funcionalidades. Mas tais assuntos são tratados sempre enfocando a adaptação do computador com o homem visando melhor servi-lo.

Deve-se considerar o tipo de usuário que utilizará o sistema proposto em cada ocasião. Para esta pesquisa, é considerada uma pessoa com conhecimento básico nas áreas de proteção, simulação e ensaios. Busca-se no projeto descrito apresentar uma interface que estabeleça com esse usuário uma relação amigável onde o mesmo possa empregar as funções do aplicativo de maneira intuitiva e confortável a partir de seus conhecimentos na área citada. Barfield [13] afirma que projetar uma interface é muito mais que projetar telas e ícones agradáveis. É uma área vital e complexa envolvendo segurança, eficiência e conforto individual

do usuário.

O desenvolvimento desse aplicativo de *software* visa a auxiliar o usuário com o perfil descrito no parágrafo anterior, em seu trabalho. Para tanto, o produto deve ter uma boa usabilidade e, segundo Nielsen[7], para que um *software* apresente usabilidade, ele deve apresentar cinco características fundamentais:

Aprendizagem: O usuário deve ter facilidade em aprender a utilizar os recursos do sistema, de forma que o mesmo possa rapidamente obter os resultados desejados.

Eficiência: O sistema deve ser eficiente fornecendo um alto grau de produtividade a um usuário que tenha aprendido a utilizá-lo.

Memorização: O sistema não deve ser complexo de forma que um usuário ficando algum tempo sem empregar tal sistema não necessite re-aprender a utilizá-lo.

Erros: O sistema deve apresentar pouca taxa de erros a fim de fornecer confiança ao usuário perante os resultados obtidos, e quando um erro é cometido, suas consequências não devem ser críticas.

Satisfação: O sistema deve fornecer satisfação ao usuário, ou seja, deve ser agradável quando utilizado.

2.2 *Cognitive Walkthrough* (Percurso Cognitivo)

O percurso cognitivo é uma técnica de avaliação de interfaces onde uma das principais preocupações é analisar o quanto um determinado sistema consegue guiar um usuário não treinado de forma que este consiga atingir seus objetivos.

A motivação para se utilizar esse método reside no fato de que a maioria dos usuários prefere aprender a utilizar um *software* por exploração, ou seja, ao invés de investir seu tempo lendo material de apoio, ou em treinamentos formais, ele prefere ir adquirindo conhecimento das características do *software* enquanto trabalha com as tarefas mais simples e usuais e, à medida que necessite, vai aprendendo a utilizar outras funcionalidades [14].

O método de avaliação percurso cognitivo não envolve os usuários e não requer um protótipo totalmente funcional [15], o que torna fácil de ser aplicado permitindo que equipes de *designers* de interfaces consigam avaliar seus aplicativos rapidamente. Ele permite que o desenvolvedor mude de perspectiva, passando a ser um usuário em potencial e, portanto, comece a identificar alguns problemas que podem surgir na interação com o sistema.

Existem alguns pré-requisitos para que o método seja corretamente utilizado [16], os quais serão citados a seguir.

Uma descrição do sistema, ou um protótipo, de forma que seja possível fazer uma navegação completa;

Uma descrição específica de tarefas ou funcionalidades que um usuário pode executar nesse sistema;

Uma lista das ações necessárias para concluir as tarefas descritas anteriormente;

Uma descrição referente ao tipo de usuário que costuma utilizar o sistema, juntamente com a sua experiência em *softwares* semelhantes e conhecimentos técnicos.

Após atingir esses pré-requisitos referenciados, o método de avaliação cognitiva propõe que esse utilizador sem qualquer conhecimento prévio, cujas características foram levantadas anteriormente, execute através do protótipo as diferentes tarefas do sistema [17]. Em [18] os autores descrevem na sua teoria de aprendizado exploratório o processo de resolução de problemas, e nela eles explicam que o usuário deve receber uma descrição aproximada da tarefa que deseja realizar, explorar a interface gráfica selecionando as ações que ele acredita que vão levar ao cumprimento da tarefa proposta. As reações que a interface apresenta para cada ação devem ser verificadas, para que o usuário saiba que ações deverão ser tomadas em seguida.

Durante a execução destes passos, o desenvolvedor que está fazendo a análise registra as respostas para algumas questões básicas que serão apresentadas a seguir [6]:

O usuário tentará atingir a meta correta?

Dada a decomposição de uma tarefa em sub-tarefas, o usuário saberá por onde começar? Saberá qual é o próximo passo?

O que o usuário vai tentar fazer a cada momento?

O usuário perceberá que a ação correta está disponível?

Onde está o elemento de interface correspondente ao próximo passo?

Que ações a interface torna disponíveis?

O usuário associará o elemento de interface correto à meta a ser atingida?

O elemento de interface revela seu propósito e comportamento?

O usuário consegue identificar os elementos de interface?

Se a ação correta é tomada, o usuário perceberá que progrediu em direção à solução da tarefa?

Como a interface apresenta o resultado de cada ação?

O resultado apresentado tem correspondência com o objetivo do usuário?

Outras perguntas podem ser feitas de acordo com o sistema, tendo em vista sempre analisar o quanto o usuário consegue aprender enquanto utiliza o aplicativo em estudo.

Após essas etapas é preciso interpretar os resultados identificando as falhas na interface gráfica atribuindo uma taxa de gravidade para cada problema detectado, e finalmente é preciso fazer um *redesign* corrigindo tais falhas.

2.3 O padrão COMTRADE

Nesta seção, com base na norma IEEE incluída nas referências dessa monografia [19], será explanada a estrutura geral do padrão IEEE COMTRADE.

O padrão IEEE COMTRADE (Common Format for Transient Data Exchange) define um formato comum para arquivo de dados digitais e meio de troca para intercâmbio de vários tipos de faltas, testes e simulações. Cada registro COMTRADE possui um conjunto de até quatro arquivos associados.

Cada um dos quatro arquivos carrega diferentes classes de informações e são apresentados a seguir:

Header;

Configuration;

Data;

Information.

Detalhes especificando cada uma destas classes de informações podem ser encontrados em [19].

Todos os arquivos do conjunto devem possuir o mesmo nome, diferindo apenas pela extensão que, neste caso, indica um dos tipos de arquivo listados acima.

Os nomes dos arquivos devem estar no formato xxxxxxxx.yyy. A porção xxxxxxxx (no máximo oito caracteres) é o nome usado para identificar o registro. Já a porção .yyy (exatamente três caracteres), a extensão, é utilizada para identificar o tipo de arquivo representado; é uma sigla associada a cada um dos tipos de arquivo: *hdr*, *cfg*, *dat* e *inf*. Estas siglas são referentes aos arquivos *Header*, *Configuration*, *Data* e *Information* respectivamente.

2.4 Aplicativos Existentes

E. P. Lenz e J. G. RolimSegundo[20] desenvolveram uma aplicação no *software* MATLAB para avaliar o desempenho de relés de proteção para geradores. Esta aplicação visa observar o comportamento de relés perante faltas/defeitos, e possui sete funções que são a função diferencial (código ANSI 87), de impedância, sobrecorrente instantânea, sobretensão, reversão de potência, perda de campo e sequência negativa. As dificuldades encontradas para o desenvolvimento do mesmo em alguns aspectos se assemelham as encontradas para o desenvolvimento do WAPS, pois ambos partiram da leitura de arquivo de oscilografia para depois realizar a análise através de gráficos. A Figura 2 mostra como os gráficos foram apresentados nessa aplicação.

TOP (*The Output Processor*) é um *software* para visualização de dados de acompanhamento e resultados de simulação[21]. Esse aplicativo foi desen-

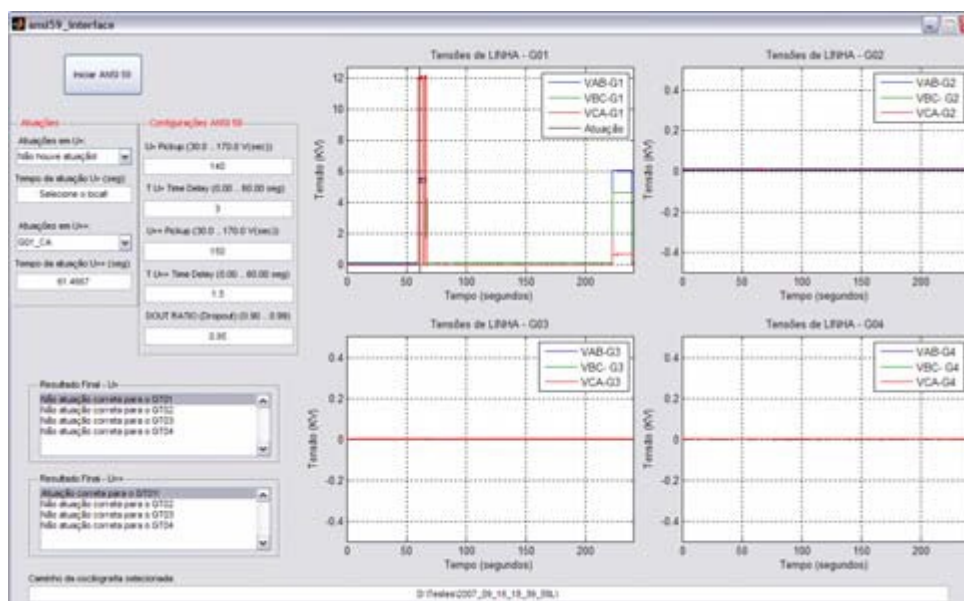


Figura 2: Interface do sistema aplicado a uma usina com quatro geradores.

volvido pela Electrotek Concepts® lendo uma grande variedade de fontes e transformando-os em gráficos de alta qualidade. Possui uma grande diversidade de gráficos, utiliza MDI, permite a visualização de várias curvas em um único gráfico, permite a realização de diversas operações matemáticas como às operações básicas, a FFT e a IDFT (*Inverse Discrete Fourier Transform*), raiz quadrada, entre outras. Contudo, não possui um cursor integrado entre os gráficos, o que dificulta a comparação dos valores entre eles. É disponível de forma gratuita e atualmente está na versão 6.02. A Figura 3 mostra a interface gráfica do aplicativo citado.

SIGRA - Powerful Analysis of all Protection Fault Records: Esse produto foi aprovado por pessoas que têm bastante experiência prática na área de avaliação de falhas [22]. Ele exibe os valores dos sinais no tempo, diagramas fasoriais, diagrama de barras para mostrar os componentes harmônicos após o cálculo da FFT. Esse produto calcula as componentes simétricas, impedâncias, saídas, valor RMS, além de possuir dois cursores para avaliação de falta em um determinado registro. Uma característica muito interessante é que o software apresenta uma interface com a capacidade de trabalhar com várias janelas internas (MDI), e os cursores são sincronizados entre todas as janelas. O *software* possui uma poderosa ferramenta de *zoom*, uma configuração bastante amigável com mecanismos de arrastar e soltar (*drag & drop*) além de ser muito eficiente. É uma ferra-

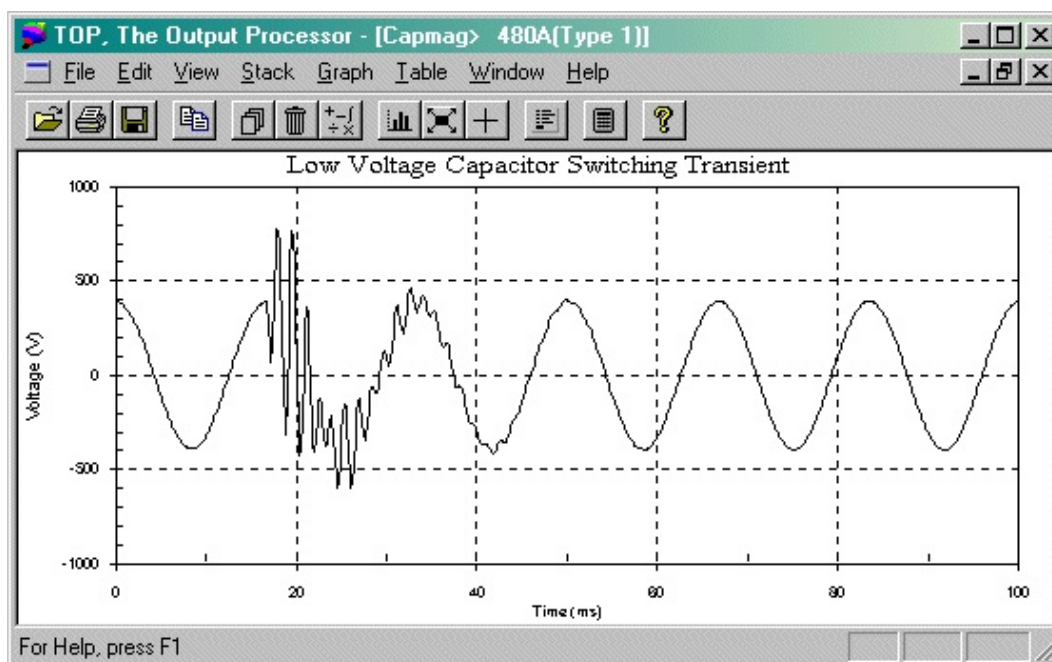


Figura 3: Interface gráfica do sistema TOP.

menta disponível apenas para o sistema operacional Windows e sua licença não é gratuita, lê arquivos apenas no formato COMTRADE. A Figura 4 mostra a interface gráfica do aplicativo citado.

.NET Relay: É um *software* bastante simples produzido em 2006[23]. Anteriormente era denominado por *Waveform Analyzers 1.0 (July 2006)*. Sua principal funcionalidade é a simulação de algoritmos e experimentos escritos pelo próprio usuário, mas também é um poderoso visualizador de arquivos COMTRADE. O software exibe os sinais das amostras em função do tempo, possui cursores e um painel bem completo e informativo para configurar as séries exibidas. Os gráficos são separados em abas diferentes, porém vários canais podem ser exibidos em um único gráfico. Esse produto tem uma funcionalidade bastante interessante de exportar para o Excel os dados da amostra. A licença desse *software* é livre, ou seja, o produto pode ser utilizado para estudos em universidades e empresas. Não possui gráficos circulares e gráficos de barra para a exibição dos valores fasoriais e dos harmônicos da FFT. Pelo *software*, os cálculos matemáticos precisam ser programados antes de serem exibidos, o que pode ser uma vantagem para cálculos muito específicos. Porém, um problema no caso de manipulações muito usuais como, por exemplo, o cálculo do valor eficaz e as operações matemáticas básicas. Observe na Figura 5 a interface do aplicativo

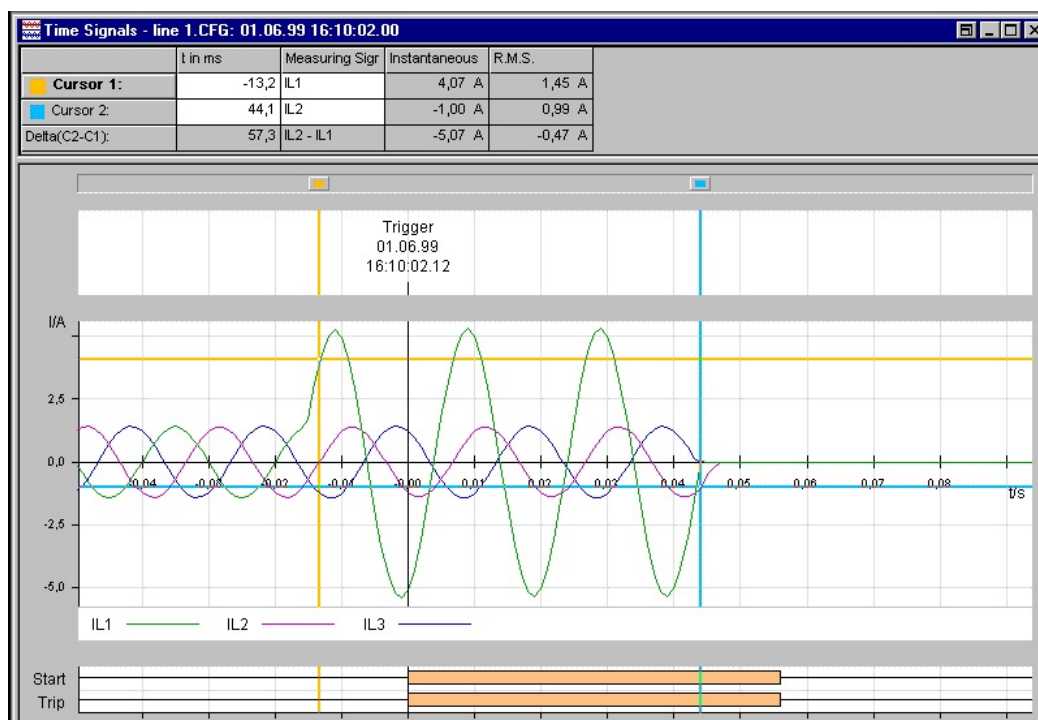


Figura 4: Interface gráfica do sistema SIGRA.

.NET Relayer.

SINAPE (Sistema Integrado de Apoio à Análise de Perturbações): Este projeto nasceu buscando suprir as mesmas necessidades de mercado apontadas na seção 1.3, visto que desde a década de 80 havia a necessidade no mercado de um *software* específico para a visualização de perturbações e oscilografia na área de proteção. Em 1995 lançaram a versão “1.0a” atendendo as principais necessidades das empresas brasileiras interessadas. O *software* continua crescendo e se mostra bem completo[24]. Atualmente está na versão 3.0. Em termos de visualização mostrou-se eficiente, pois permite à comparação de oscilografia das mais variadas formas. Ele exhibe os valores instantâneos, fasores e eficazes, além de possuir cursores para a comparação entre gráficos. É bastante modular, o que significa que diferentes funcionalidades podem ser incorporadas ao software de acordo com o interesse das empresas. Outras funcionalidades que chamam a atenção são a sincronização entre ondas, visualização de canais de diferentes estações, análise harmônica, localização de falta, gráficos de impedância e exportação em formato COMTRADE. Porém esse software não é livre e, portanto, não pode ser usado para o estudo sem uma licença. Apenas arquivos no formato COMTRADE são importados e exportados. A Figura 6 mostra a interface gráfica do SINAPE.

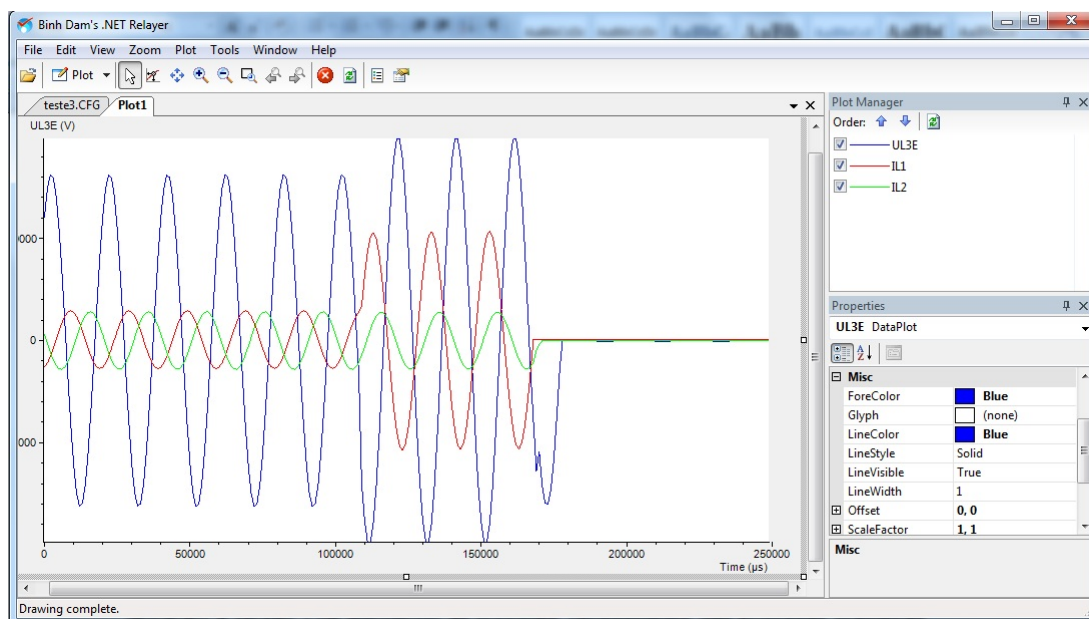


Figura 5: Interface gráfica do sistema *.NET Relayer*.

Para o desenvolvimento do trabalho em questão, foram consideradas algumas características dos aplicativos descritos neste capítulo. Espera-se que o novo aplicativo tenha uma tabela para a configuração de séries com funcionalidades semelhantes ao *.NET Relayer*. Espera-se também que ele tenha a possibilidade de manipular diversos arquivos e formatos conforme o TOP, trabalhe com muito gráficos simultâneos e cursores conforme é o SIGRA, trabalhe com MDI, seja um *software* livre que exiba várias curvas em um único gráfico.

Todas essas características são encontradas nos aplicativos descritos acima. Porém esse conjunto não é encontrado em um único aplicativo. Assim detectou-se a necessidade da criação do WAPS para reunir essas características.

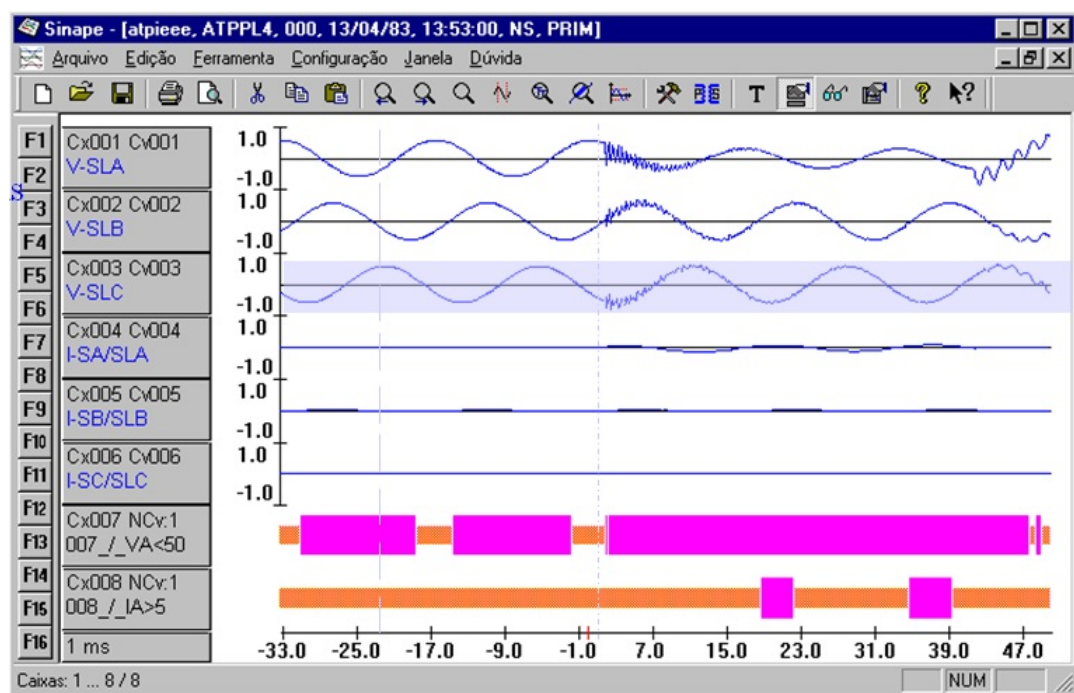


Figura 6: Interface gráfica do sistema SINAPE.

3 A Ferramenta WAPS

Conforme descrito anteriormente, a ferramenta WAPS está dividida em dois módulos que terão suas funcionalidades descritas nas próximas seções.

3.1 Descrição do Módulo 1

Conforme já mencionado, existem diversos formatos de dados responsáveis por armazenar valores relativos aos transitórios e as perturbações em redes de energia elétrica. Podemos citar o IEEE *Standard Common Format For Transient Data Exchange For Power System* (COMTRADE) e o *Power Quality Data Interchange Format* (PQDIF), dentre outros. O módulo 1 deverá ser capaz de ler estes formatos e transformá-los em um formato comum de dados ao WAPS.

O formato comum do WAPS dividirá esses dados em canais conforme o esquema mostrado na Figura 7.

Observando a figura, nota-se que os dados brutos serão divididos em canais que poderão ser analógicos, digitais e derivados. O canal analógico representará os dados de um sinal contínuo no tempo que foi discretizado para a representação computacional. O digital por sua vez, também representa um sinal contínuo no tempo, porém binário, contendo apenas os bits 0 e 1. Também observam-se os derivados, que são canais gerados pelo próprio WAPS. Esses canais estão divididos em duas subcategorias que são os aritméticos, gerados a partir de operações algébricas sobre os canais analógicos, e FFTs, gerados a partir de uma transformada discreta de Fourier utilizando o algoritmo FFT sobre canais analógicos.

Funcionamento do módulo 1:

Abrir arquivo e selecionar canais.

O usuário deverá carregar os arquivos que deseja utilizar. Para tanto ele

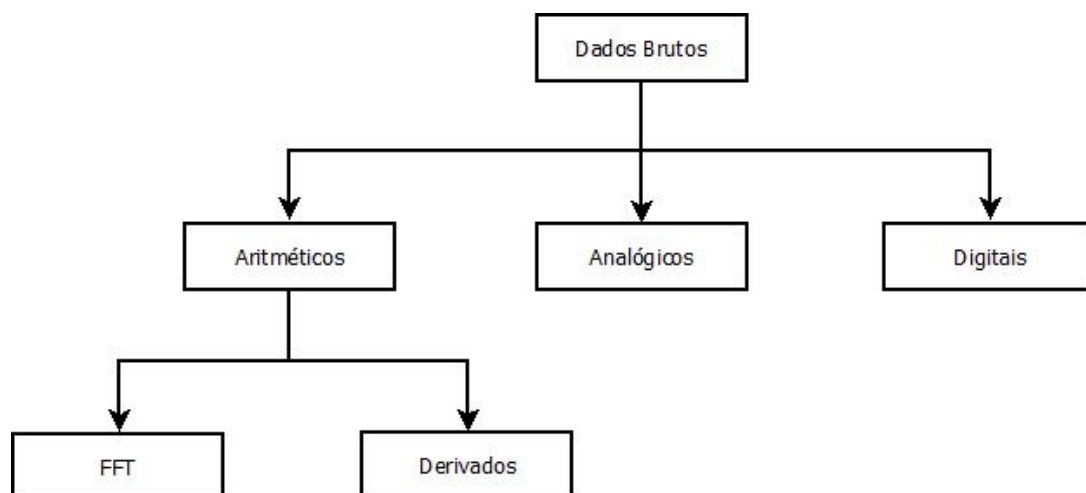


Figura 7: Esquema de divisão de canais do WAPS.

deverá clicar no ícone de abrir na barra de tarefas, ou em abrir no menu arquivo. A Figura 8 mostra a tela que é exibida assim que usuário realizar tal ação na opção de abrir. Essa tela é conhecida como seletor de arquivos, comum em muitos *softwares*. Ela permite que o usuário selecione arquivos. No caso do WAPS precisa ser um arquivo de configuração do padrão COMTRADE (extensão “.cfg”).

Assim que carregar um arquivo, o usuário poderá escolher quais canais deseja trabalhar. A Figura 9 mostra a interface disponível ao usuário. Essa tela ao longo do documento será referenciada como janela de seleção de canais. Nela o usuário deve selecionar os canais analógicos e digitais que deseja trabalhar e clicar no botão “OK”. Para a categoria de canais analógicos existe um filtro referente a unidade de cada canal. Dessa forma é possível visualizar apenas os canais das seguintes unidades:

Tensão: em Volts (V);

Corrente: em Ampéres (A);

Potência: em Watts (W);

Outros: qualquer canal com unidade diferente das mencionadas.

Temos ainda o botão “Selecionar Todos” que permite a seleção de todos os canais de uma determinada categoria e o botão “Nenhum” que tira a seleção de todos os canais de uma categoria de canal.

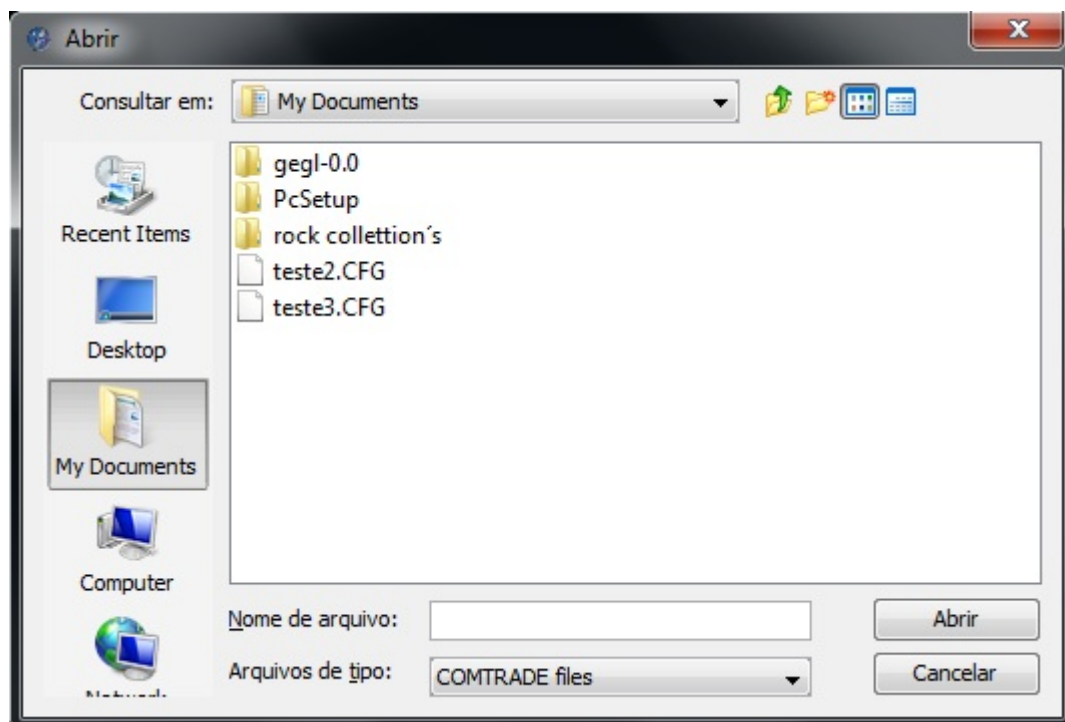


Figura 8: Seletor de arquivos de configuração COMTRADE.

Math Builder, a ferramenta matemática do WAPS

Depois de escolher os arquivos e canais que deseja trabalhar, o usuário poderá realizar operações algébricas sobre/entre os canais analógicos, cálculo da FFT e da IDFT através da ferramenta matemática do WAPS. Essa ferramenta foi desenvolvida de maneira que outros algoritmos possam ser incorporados ao sistema sem grandes alterações. A Figura 10 mostra sua interface. Assim que a operação é realizada, um novo canal é criado, e conforme mencionado, este canal será tratado como canal derivado.

Para utilizar essa ferramenta, o usuário deverá primeiramente selecionar o canal sobre o qual será aplicada alguma operação. Deve-se considerar que algumas operações são restritas a uma determinada categoria de canais.

Após selecionar o canal o usuário poderá escolher a operação que deseja realizar nele, podendo ser uma das quatro operações básicas (soma, subtração, multiplicação e divisão). Nesse caso é preciso selecionar o outro argumento da operação, que pode ser uma constante ou uma curva. Outra possibilidade seria realizar as operações auxiliares que são a operação de inversão ($1/x$) e negatização ($-x$), sendo que essas operações não necessitam do segundo argumento. Essas

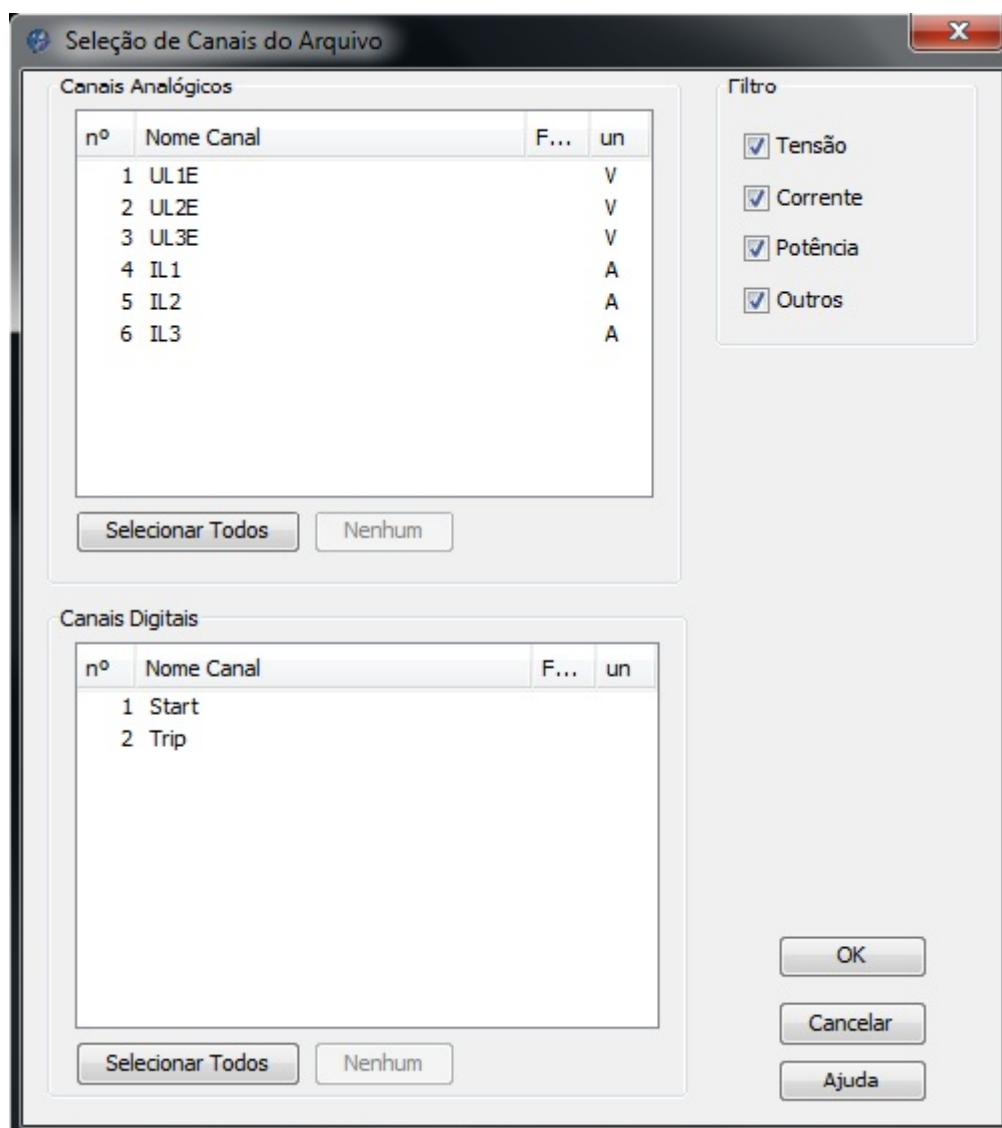


Figura 9: Janela de Seleção de Canais de um arquivo.

operações podem ser feitas entre canais analógicos e canais derivados aritméticos.

O *software* também possibilita a realização de operações de transformadas (FFT e IDFT), que utilizam um novo domínio de referência diferente do tempo. Este domínio, denominado domínio da frequência, representa seus sinais em componentes espectrais e por isso existem algumas restrições de operações conforme o tipo de canal. Um novo menu é disponibilizado (menu “Parâmetros”) para a configuração de alguns parâmetros necessários para a transformação de um sinal no domínio do tempo para o domínio da frequência.

Menu Parâmetros

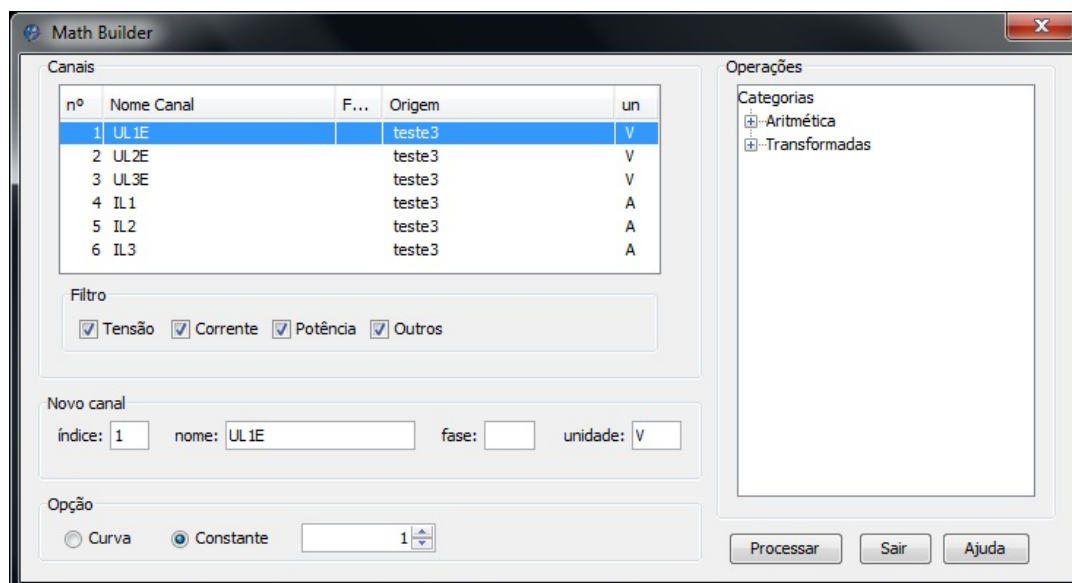


Figura 10: Interface da ferramenta matemática do WAPS.

A operação FFT disponibiliza um menu de parâmetros para a configuração de algumas propriedades:

Frequência Fundamental(Hz): Valor da frequência (em Hertz) fundamental base para as harmônicas múltiplas. É o valor da frequência do sistema fornecido no arquivo de configuração.

Tempo inicial(ms): Tempo (em milissegundos) do começo da transformação. Dados obtidos antes desse tempo serão desconsiderados. O valor inicial é 0.

Número de ciclos: Número de ciclos da amostra. O valor inicial é 1. Caso seja fornecido um valor maior que o número máximo de ciclos da amostra então será adotado o valor máximo para o cálculo.

A operação escolhida será aplicada sobre um canal sem contudo modificá-lo. Para isso um novo canal será gerado e, portanto, torna-se necessário a definição de um novo canal com os dados gerados a partir do resultado obtido.

As seguintes informações devem ser inseridas para a criação do novo canal:

índice: número do canal;

nome: nome de identificação do canal;

fase: fase do canal;

unidade: unidade de medida do canal.

Após clicar em “Processar” para criar o novo canal, o resultado da operação realizada será apresentado para o usuário.

Configurar exibição de gráficos

Depois de ter carregado e criado todos os canais, o usuário poderá enfim decidir de que maneira irá visualizar os gráficos. Para isso houve a necessidade da criação de mais uma tela para a divisão dos canais nas áreas de impressão. Essa tela será chamada de janela de configuração para a exibição de gráficos, e nela o usuário poderá visualizar todos os canais, separados por tipo (analógico, digital e derivado), e os transferir para uma das quatro áreas de impressão. Cada área representa um painel de exibição de gráfico distinto, ou seja, este programa exibirá no máximo quatro painéis de exibição de gráficos, porém um gráfico poderá ter mais de uma curva. Observe a interface dessa tela na Figura 11.

Os canais devem ser selecionados e transferidos para a área desejada. Podem-se selecionar vários canais simultaneamente através do mouse (botão esquerdo) e pelo teclado (tecla Ctrl).

Existem 3 formas de transferência de canais:

Arrastar e soltar: Após selecionados os canais de interesse, basta pressionar o botão esquerdo do mouse sobre um dos itens da seleção e mantê-lo pressionado enquanto arrasta-se o mouse até uma das áreas, soltando-se em seguida o botão pressionado.

Botões: O botão “Mover Selecionados” permite que os canais previamente selecionados sejam movidos automaticamente para a área que estiver selecionada. O botão “Mover Todos” move todos os canais da palheta em destaque para a área selecionada independente de qualquer seleção previamente realizada.

Imagem de transferência: Os canais podem ser movidos individualmente através da imagem ao final de cada linha da palheta que representa a transferência para a área que estiver em destaque.

Assim que escolhidos os canais, também ficou a cargo do módulo 1, calcular os

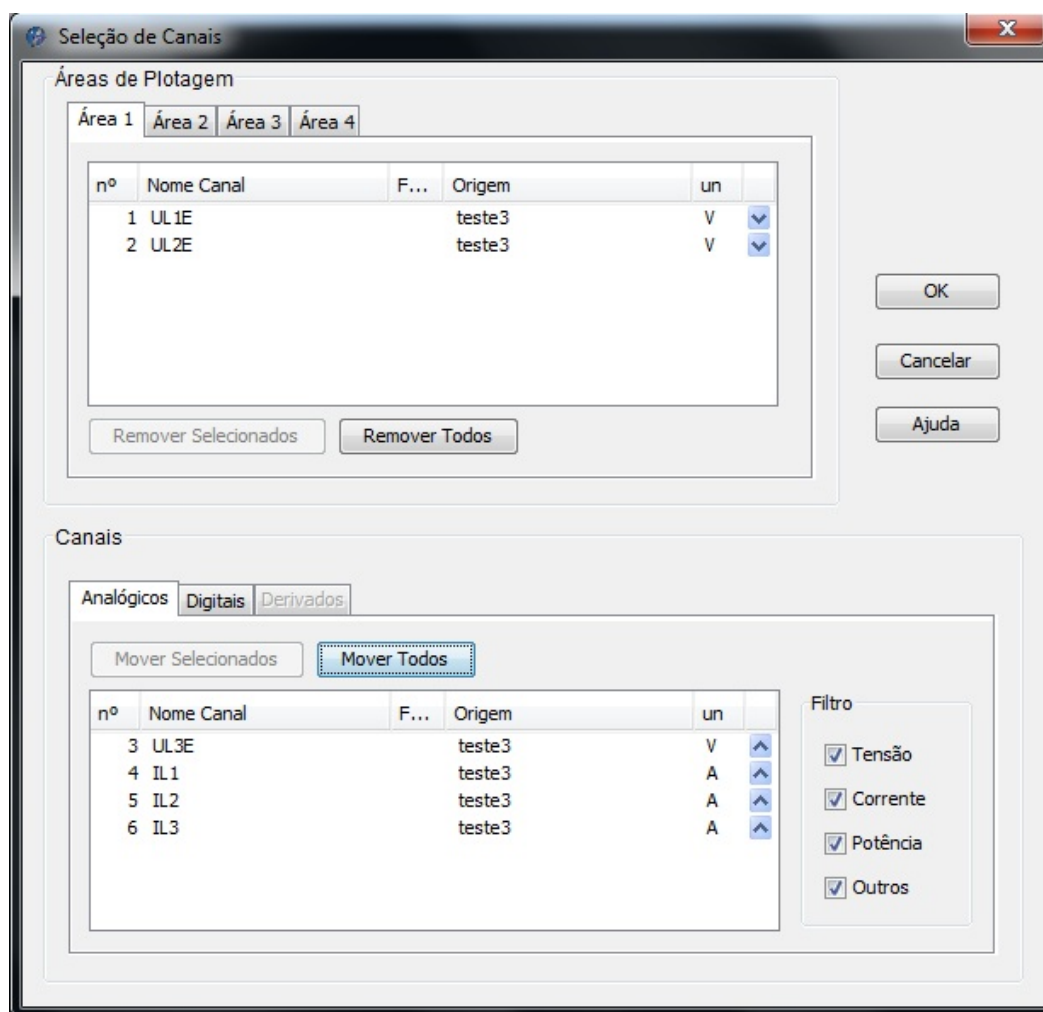


Figura 11: Janela de Configuração para a Exibição de Gráficos.

valores RMS e fasoriais de cada ponto das curvas dos canais analógicos e derivados aritméticos. Esses valores foram salvos em estruturas para posteriormente serem utilizados pelo módulo 2.

Tela de Descarte

A tela de descarte provê a desabilitação de canais contidos no escopo de visibilidade do sistema.

Os canais são divididos de acordo com as categorias de canais existentes no sistema (conforme citado anteriormente). Desta forma, para se descartar um canal basta selecionar a categoria a qual ele pertence, clicar sobre o canal (ou selecionar um conjunto de canais) e escolher a opção “Descartar”. A Figura 12.

Com todas essas características citadas descrevendo o módulo 1, têm-se um

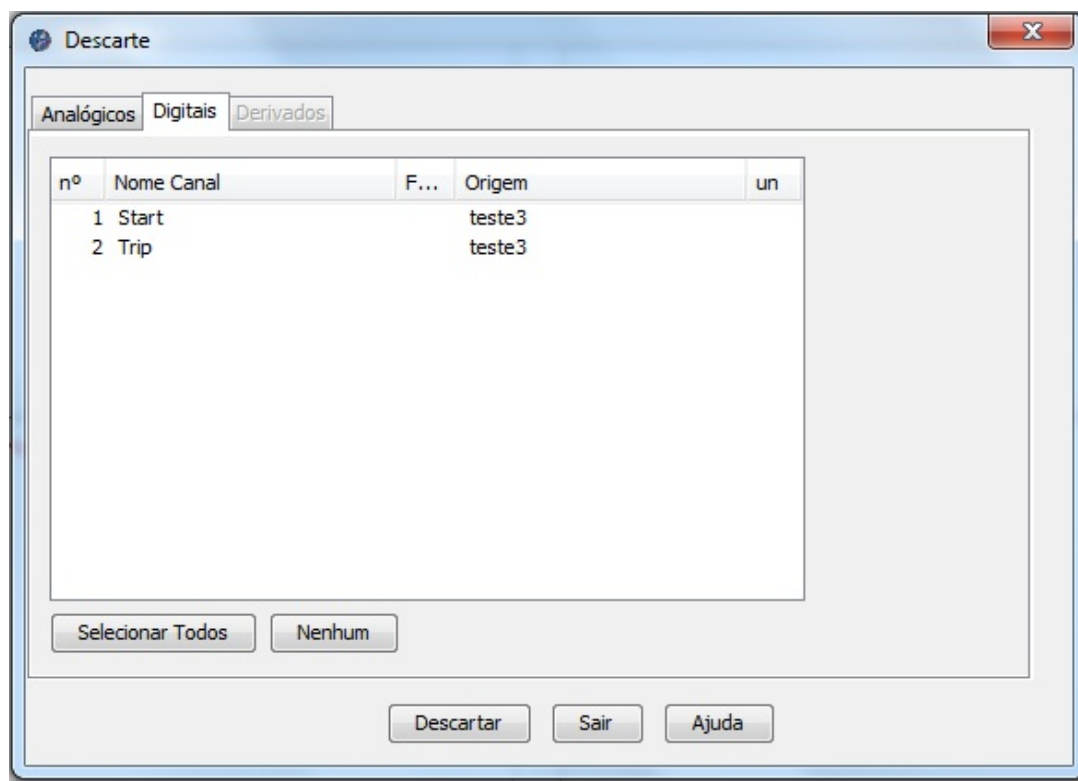


Figura 12: Interface Gráfica da Janela de Descarte.

ambiente favorável ao desenvolvimento do módulo 2. A seguir será descrito o funcionamento do módulo 2.

3.2 Descrição do Módulo 2

Esse módulo tem como objetivo ler os dados dos canais criados e carregados pelo módulo 1 e exibi-los ao usuário através de gráficos. Esses gráficos deverão respeitar as grandezas e unidades de cada canal, exibir as curvas com clareza, permitindo o usuário colocar mais de uma curva em um único gráfico. Existe a necessidade de uma ferramenta de *zoom* para mostrar com maior precisão uma curva.

Além disso, em caso de gráficos analógicos, digitais e aritméticos, dois cursores serão utilizados para percorrer a curva possibilitando a obtenção da magnitude da mesma em um determinado ponto. Para o aplicativo foi detectado a necessidade de dois cursores para melhorar as possibilidades de comparação entre curvas. Dessa forma o WAPS deverá apresentar através de uma tabela a soma, divisão,

subtração e multiplicação dos valores apontados pelos cursores.

Conforme mencionado na introdução, são de grande importância a um profissional da área de proteção os valores RMS e fasoriais de um transitório. Assim o módulo dois deverá utilizar os valores calculados no módulo 1 e exibi-los também através de gráficos. Para aumentar as possibilidades de comparação e conforto na utilização desse aplicativo, as janelas serão exibidas por *Multiple Document Interface* (MDI) que é uma maneira onde o usuário tem a possibilidade de trabalhar com várias janelas filho dentro de uma janela pai.

Tela de exibição de gráficos

Os gráficos são apresentados em até quatro painéis, sendo que cada painel representa uma das áreas de exibição do sistema. Note que o número de painéis varia de acordo com o número de áreas preenchidas pelo usuário quando o mesmo configurou a exibição dos gráficos (janela de configuração para a exibição de gráficos). Para cada canal colocado em uma área de exibição haverá uma série correspondente traçada no gráfico. Ou seja, tem-se uma série representando cada canal. Além dos gráficos traçados existe, na parte superior dessa janela, um painel de opções.

A Figura 13 mostra seis canais analógicos divididos entre as áreas 1 e 2, e dois canais digitais exibidos na área 3. Pode-se observar nessa tela de exibição de gráficos, nas laterais da área de impressão, a existência de diferentes unidades de medidas. Na parte inferior tem-se uma legenda com a cor de cada curva, juntamente com sua respectiva unidade de medida. Observa-se também na figura, duas linhas verticais posicionadas dentro do painel de exibição de gráficos. Essas linhas verticais serão referenciadas nesse texto como cursores.

Zoom

Para melhor visualizar os gráficos no painel interno de exibição de gráficos, pode-se aplicar um zoom nos mesmos, independente do tipo de gráfico exibido. Para tanto, alguns pontos devem ser observados.

No caso de gráficos analógicos, digitais, fasoriais e derivados aritméticos, basta clicar com o botão esquerdo do mouse em um ponto qualquer do gráfico e, mantendo-o pressionado, arrastá-lo, soltando-o em um ponto à direita e abaixo do ponto clicado anteriormente. No caso de gráficos derivados do tipo FFT, o

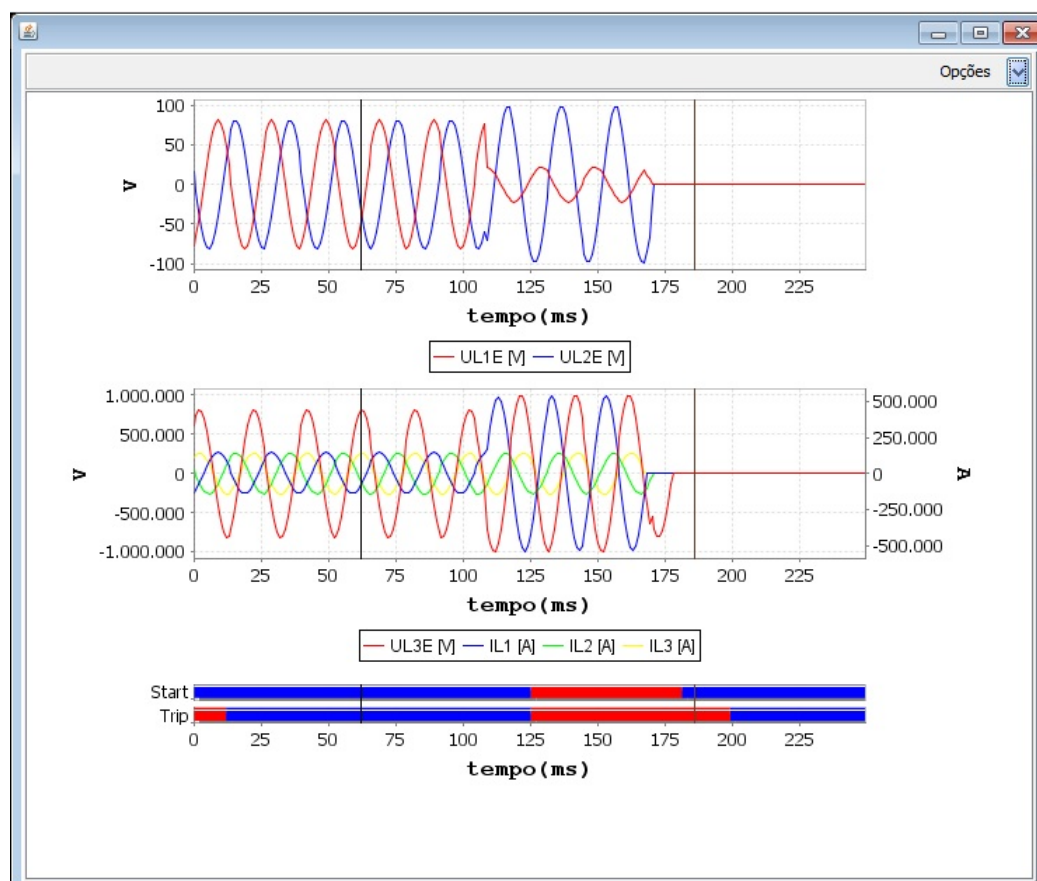


Figura 13: Janela de exibição de gráficos.

ponto onde se solta o clique do mouse deve ser apenas abaixo da posição onde aconteceu o clique do mouse.

Para que os gráficos retornem ao zoom original, o mesmo procedimento deve ser seguido, porém o clique do mouse deve ser solto em um ponto acima da posição onde o primeiro clique foi realizado.

Painel de Opções

O painel de opções pode ser observado na Figura 14. Ele é habilitado e desabilitado clicando no botão localizado na parte superior direita da janela interna de exibição de gráficos. Ele está dividido em três partes: Tabela de Operações, Tabela de Tempo e Painel Tabulado.

Tabela de Operações

Essa tabela apresenta seis linhas e três colunas. As linhas da coluna 1 apresentam o valor (magnitude) dos cursores 1 e 2, juntamente com as quatro operações

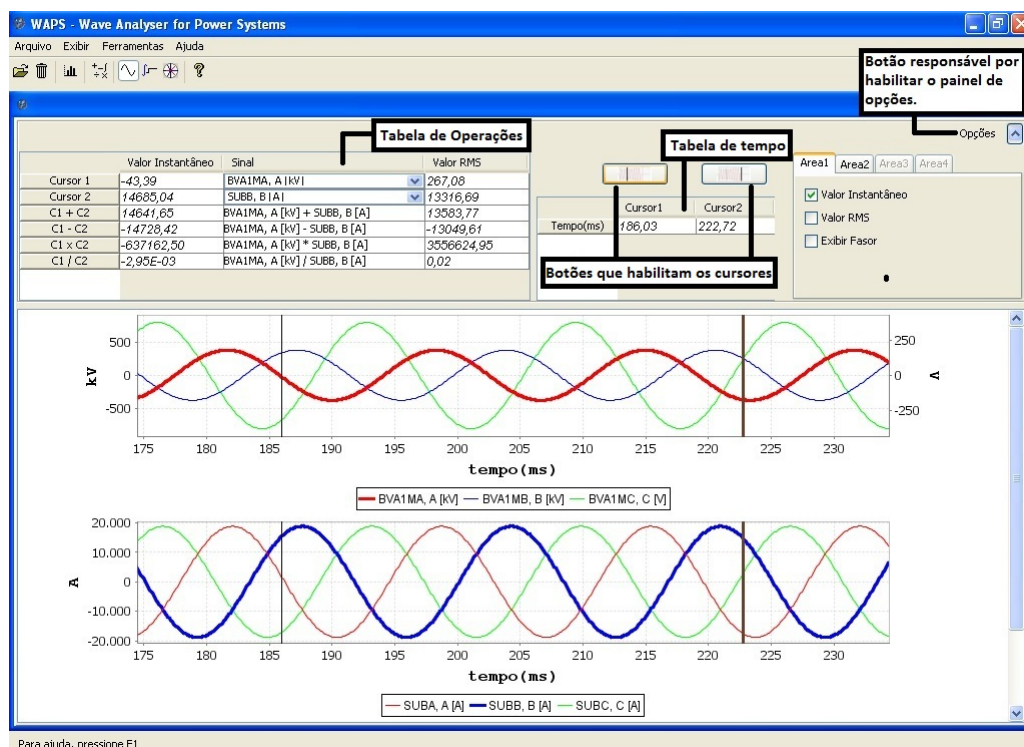


Figura 14: Interface gráfica do painel de opções.

aritméticas básicas entre eles em um determinado instante de tempo. As duas primeiras linhas da segunda coluna apresentam caixas de combinação com os nomes de todas as séries analógicas e derivadas aritméticas plotadas nos gráficos. Enquanto nenhum canal é escolhido, as operações não são realizadas e, portanto, a tabela mantém os seus valores em zero. Caso não haja canais analógicos ou aritméticos, a tabela não atualizará seus valores. Quando um canal é selecionado, o traço no gráfico fica mais largo para facilitar sua visualização.

Por fim, na última coluna da tabela, encontram-se os valores em um determinado instante de tempo dos cursores 1 e 2 referente ao sinal RMS da série selecionada nas caixas de combinação. Analogamente a primeira coluna, abaixo das linhas onde se encontra os valores dos cursores 1 e 2, observa-se as quatro operações aritméticas básicas entre os cursores.

Tabela de Tempo (Figura 14)

A tabela apresenta duas células, sendo elas responsáveis por mostrar os tempos em milissegundos sobre os quais os cursores passam. Além da tabela de tempo, encontra-se logo acima os botões "Ativar Cursor 1" e "Ativar Cursor 2",

sendo cada um deles responsável por ativar um dos cursores.

Painel Tabulado (Figura 14)

O painel tabulado da barra de opções apresenta 4 abas, sendo que cada aba representa um dos painéis internos dos gráficos analógicos ou derivados aritméticos traçados. Quando houver gráficos digitais ou derivados FFT, a aba referente a esse gráfico é desabilitada.

Cada aba possui três caixas de seleção, sendo que quando a primeira caixa de seleção é marcada (Valor Instantâneo), as curvas do tipo valor instantâneo passam a ser mostradas. Quando essa caixa de seleção é desmarcada, é verificado se a segunda caixa de seleção (valor RMS) está marcada. Caso a mesma esteja marcada, o gráfico passa a exibir somente as séries de valor RMS. Se desmarcada a janela interna de exibição de gráficos exibe a interface de gráficos fasoriais referente aos canais da área correspondente a aba que possui a caixa de seleção desmarcada.

Para a segunda caixa de seleção (valor RMS), quando marcada, são exibidas as séries de valor RMS dos canais traçados nos gráficos, e quando desmarcada, é verificado se a primeira caixa de seleção está selecionada. Caso marcado, as curvas RMS deixam de ser mostrados e caso desmarcada, a janela interna de exibição de gráficos passa a exibir a interface de gráficos fasoriais com os fasores das séries referentes ao gráfico cuja aba possua a caixa de seleção utilizada.

Por fim, no caso da terceira caixa de seleção, quando marcada, a interface de gráficos fasoriais é exibida com os mesmos critérios dos casos anteriores, e quando desmarcada, as séries de valor instantâneo são mostradas.

Cursores (Figura 14)

Movimentando os cursores, o usuário tem a possibilidade de visualizar a magnitude de uma curva em um determinado instante de tempo do gráfico, através da tabela de operações e da tabela de tempo no painel de opções.

Para movimentar os cursores, é preciso habilitá-los através dos botões “Ativar Cursor 1” e “Ativar Cursor 2” localizados na tabela de tempo dentro do painel de opções. O botão “Ativar Cursor 1” habilita o cursor 1 e desabilita o cursor 2, e em contrapartida o botão “Ativar Cursor 2” habilita o cursor 2 e desabilita o cursor 1.

Quando os cursores não estão habilitados, a única maneira de movimenta-los é através dos controles deslizantes da interface de exibição de gráficos fasoriais. Depois de habilitado, pode-se alterar a posição do cursor de 2 maneiras:

Duplo Clique: Após a realização de um duplo clique na área de impressão, o cursor habilitado passa a ocupar a posição horizontal do *mouse*.

Clique sobre linha vertical Ao posicionar o *mouse* próximo, ou em cima de um dos cursores habilitados, nota-se que o ícone do *mouse* é modificado para cursor de redimensionamento. Realizando um clique, a linha vertical passa a acompanhar a posição horizontal do mouse na área de impressão do gráfico. Após outro clique o cursor para de acompanhá-lo tomando a posição onde ele foi realizado.

Gráficos Fasoriais

Cada canal analógico ou derivado aritmético de um determinado gráfico terá uma série representando o ângulo e a amplitude máxima em um certo tempo. Essas séries serão mostradas em uma interface da própria janela interna de exibição de gráficos, sendo que esta possui o painel de opções, dois painéis de exibição de gráficos e dois controles deslizantes (um painel e um controle deslizante a direita e o outro painel e controle deslizante a esquerda). Os controles deslizantes determinam o tempo que será representado o ângulo e a fase das séries nos gráficos. A Figura 15 mostra a interface descrita.

Encontra-se nos painéis de exibição de gráficos um gráfico do tipo vetorial. Este apresenta um vetor correspondente a cada série. O vetor é a projeção bidimensional do sinal no tempo que é indicado pelo controle deslizante. Na ponta deste se verifica a amplitude e a fase da série. Conforme o controle deslizante da esquerda se movimenta, o gráfico do painel da esquerda é modificado, sendo análogo para o controle deslizante da direita e o gráfico do painel da direita.

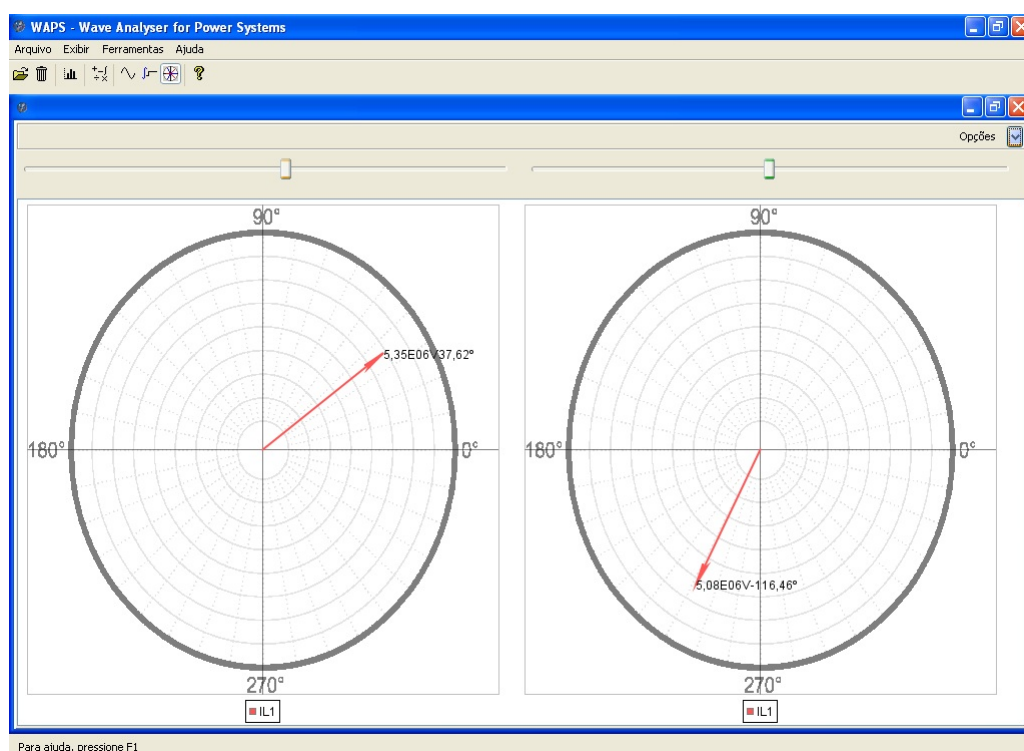


Figura 15: Janela de exibição de gráficos fasoriais.

4 Metodologia de Trabalho

4.1 Preparação do Percurso Cognitivo

O método do percurso cognitivo é composto de duas fases, sendo a primeira preparatória e a segunda analítica. Nesta seção da metodologia serão definidas as entradas para o método, ou seja, as tarefas, as sequências de ações para cada tarefa, os usuários característicos e a interface que será submetida ao método.

Que tipo de usuário utiliza o sistema WAPS?

Os desenvolvedores desse projeto esperam que três grupos de usuários utilizem o sistema. A seguir serão descritas as características de cada grupo. Note que essas características foram levantadas pelos próprios desenvolvedores a partir da experiência deles na área de SEP [25].

Engenheiros e Analistas de uma empresa da área: Esses usuários costumam ter bastante conhecimento técnico e prático dos assuntos dessa área, têm experiência com *softwares* semelhantes ao WAPS, além de experiência com aplicativos para escritório.

Pesquisadores da área de sistemas elétricos de potência: Esse grupo tem conhecimento técnico suficiente para a utilização do *software* e conhecimento prático um pouco menor que o grupo anterior. Acredita-se que os pesquisadores tenham acesso a uma diversidade maior de aplicativos e interfaces gráficas diferentes, pois estão diretamente ligados a universidades, ao ensino e a pesquisa.

Alunos de graduação dos cursos de engenharia elétrica e engenharia de computação: Da mesma forma que o segundo grupo descrito, acredita-se que eles também tenham acesso a uma grande diversidade de

aplicativos e interfaces gráficas. Porém o conhecimento técnico e prático é um pouco mais imaturo.

Quais tarefas serão analisadas?

A seguir serão apenas citadas as tarefas definidas pelo grupo de desenvolvedores. Essas tarefas serão melhores compreendidas quando for definido o conjunto de ações que as compõe.

Tarefa 1 - Abrir arquivos e selecionar canais: Essa é a tarefa onde o usuário seleciona o arquivo e escolhe os canais para futuramente fazer qualquer tipo de análise e manipulação disponibilizada pelo aplicativo WAPS.

Tarefa 2 - Traçar gráficos: Consiste em selecionar quais canais serão exibidos através dos gráficos, escolhendo também a disposição dos mesmos nas quatro áreas de impressão.

Tarefa 3 - Movimentar cursores: Dentro do escopo dessa tarefa temos todos os passos necessários para estudar duas curvas através dos cursores. Esses passos serão descrito posteriormente, na definição da sequência de ações dessa tarefa.

Tarefa 4 - Exibir e analisar fasores: Para essa tarefa temos todas as ações relacionadas a análise de fasores, como, por exemplo, a movimentação dos cursores na tela de exibição de fasores.

Tarefa 5 - *Ferramenta Math Builder*: Será estudado nessa tarefa a interface gráfica envolvida na manipulação algébrica das curvas.

Tarefa 6 - Descartar canais: A sexta tarefa será o conjunto de ações necessárias para descartar canais, incluindo a interface gráfica da tela de descarte de canais.

A figura 16 mostra um diagrama com as tarefas descritas acima.

Qual é a sequência correta de ações para cada tarefa? A seguir será descrito o conjunto de ações envolvidas para se realizar cada uma das tarefas.

Tarefa 1 - Abrir arquivos e selecionar canais

1. Clicar na opção abrir
2. Selecionar o arquivo de configuração dos canais que o usuário pretende estudar.
3. Selecionar quais canais analógicos e digitais do arquivo serão carregados.
4. Finalizar tarefa.

Tarefa 2 - Traçar gráficos

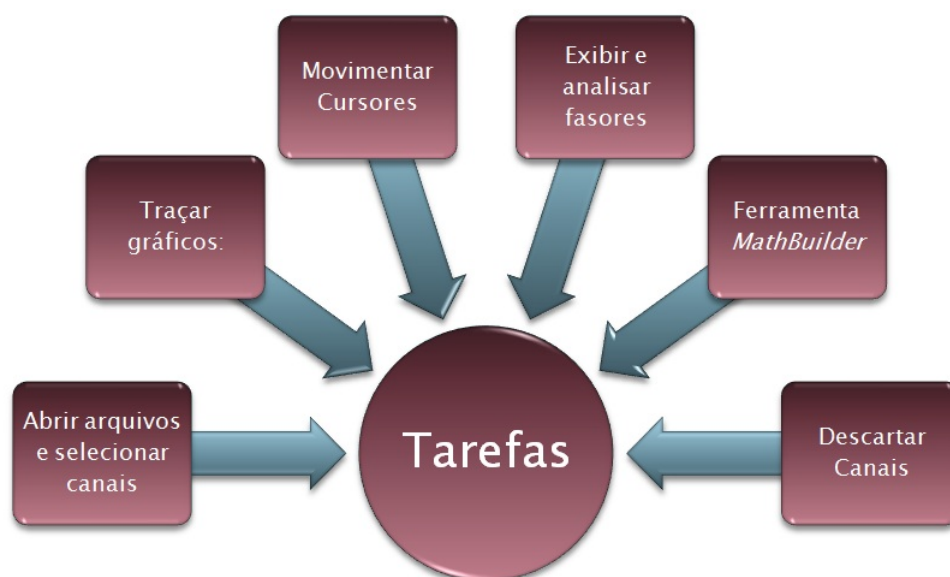


Figura 16: Diagrama representando as tarefas analisadas na aplicação do Percorso Cognitivo.

1. Clicar no botão traçar gráficos. 2. Escolher quais canais serão traçados, dividindo-os entre as quatro áreas de impressão. 3. Finalizar tarefa.

Tarefa 3 - Movimentar cursores

1. Clicar na barra de opções. 2. Habilitar um dos cursores. 3. Selecionar uma curva na tabela de operações. 4. Habilitar o outro cursor. 5. Selecionar outra curva. 6. Arrastar cursores e comparar curvas.

Tarefa 4 - Exibir e analisar fasores

1. Alternar para tela de exibição de fasores. 2. Movimentar cursores. 3. Comparar curvas.

Tarefa 5 - Ferramenta *Math Builder*

1. Clicar na ferramenta *Math Builder*. 2. Realizar operação aritmética entre curvas e constante. 3. Realizar operação aritmética entre duas curvas. 4. Aplicar a FFT em uma curva. 5. Aplicar a anti-transformada de Fourier em uma curva. 6. Sair da tela *Math Builder*.

Tarefa 6 - Descartar canais

1. Clicar na opção descartar canais. 2. Selecionar os canais que serão descartados. 3. Sair da tela de descarte de canais.

Tabela 1: Perguntas para a fase analítica do percurso cognitivo.

Perguntas	
1	O usuário saberá por onde começar? Saberá qual é o próximo passo?
2	O que o usuário vai tentar fazer a cada momento?
3	Onde está o elemento de interface correspondente ao próximo passo?
4	Que ações a interface torna disponíveis?
5	O elemento de interface revela seu propósito e comportamento?
6	O usuário consegue identificar os elementos de interface?
7	Como a interface apresenta o resultado de cada ação?
8	O resultado apresentado tem correspondência com o objetivo do usuário?

Como o sistema está descrito?

Toda a descrição desse sistema já foi anteriormente apresentada no capítulo 2.

Com isso se encerra a fase preparatória.

Contudo, antes de começar a fase analítica, ainda falta definir quais critérios de análise serão utilizados.

Conforme explicado no capítulo 3, é preciso agora que o analisador se coloque no lugar do usuário, ou seja, se coloque no lugar daquele cujas características foram definidas na fase preparatória, e execute todas as seis tarefas através das ações determinadas.

Para auxiliar na análise, para cada ação será considerada uma série de perguntas, mostradas na tabela 1.

4.2 Metodologia Incremental

Para facilitar o desenvolvimento das melhorias propostas para o aplicativo, será utilizada a mesma metodologia que serviu como base na implementação dos módulos 1 e 2 do aplicativo WAPS descritos nesse documento. Para definir esta metodologia, foram analisadas diversas metodologias de desenvolvimento de *software* considerando suas vantagens e desvantagens, possibilitando, dessa forma, escolher aquela que apresente maior ganho de produtividade, organização, documentação e padronização ao aplicativo em questão.

As metodologias estudadas foram[12]:

Convencional ou cascata;

Incremental;

Iterativo;

Espiral;

Prototipação;

RUP; e

Extreme Programming (XP).

Cabe adiantar que a metodologia que mais se adequou ao projeto em questão foi a incremental. A seguir esta será apresentada juntamente aos motivos que levaram a adoção da mesma.

O modelo incremental segundo Pressman[12] combina elementos do modelo convencional, onde o aplicativo inteiro é dividido nas fases de análise, projeto, codificação e teste, porém, de forma iterativa. O objetivo principal desse modelo é apresentar no final de cada incremento o aplicativo operando. A Figura 17 mostra as etapas do modelo incremental.

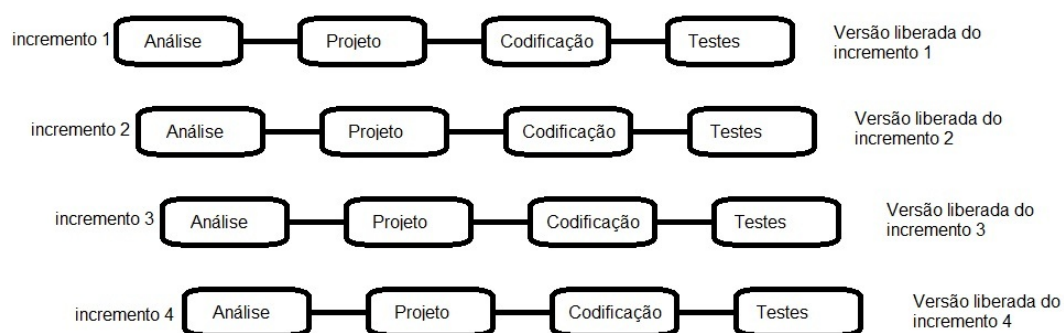


Figura 17: Diagrama representando o modelo Incremental.

Em cada ciclo são realizadas todas as etapas de desenvolvimento, e no final de cada ciclo, é apresentado o sistema totalmente funcional. Nesta fase, caso o sistema ainda não cubra todos os requisitos do projeto, as etapas são continuadas até que todos os incrementos estejam funcionando.

A construção de um incremento é menos arriscada, pois este pode ser facilmente corrigido ou descartado caso grandes erros sejam cometidos, o que não acontece na construção de sistemas grandes. Além disso, o tempo de desenvolvimento de um incremento é relativamente pequeno quando comparado com o tempo total de desenvolvimento do sistema, logo as chances de mudança de requisitos são menores.

Assim que o desenvolvedor termina um incremento, ele pode mostrá-lo para o cliente. Dessa forma o cliente pode esclarecer as dúvidas do desenvolvedor e utilizar a parte pronta do sistema, e com isso, os resultados ficam mais próximos dos resultados esperados pelo cliente.

Essas características fazem da metodologia incremental a mais adequada ao trabalho proposto, pois os requisitos deste não estão totalmente claros. Sendo assim, as melhorias do mesmo serão desenvolvidas a partir de vários incrementos, e durante sua construção, os requisitos serão definidos com maior precisão e clareza. Além disso, espera-se que, após a conclusão dos primeiros incrementos, o aplicativo seja colocado sob avaliação, possibilitando a detecção inicial de problemas que poderiam ganhar dimensões muito maiores se detectadas somente na finalização das melhorias do aplicativo.

Esta metodologia é recomendada para equipes de desenvolvimento menos experientes, pois, na ocorrência de erros, apenas o último incremento será descartado.

4.3 Ferramentas computacionais aplicadas

A seguir serão mostradas algumas das ferramentas que foram utilizadas para o desenvolvimento da interface gráfica.

A análise, projeto e desenvolvimento deste *software* seguiram o paradigma de orientação a objetos. Esse paradigma pressupõe uma organização do mesmo em coleção de objetos, que incorporam estrutura e comportamentos próprios[26].

Dizer que um *software* foi desenvolvido por uma linguagem de programação orientada a objetos significa dizer que uma série de conceitos como abstração, encapsulamento, compartilhamento, herança e polimorfismo, entre outros tantos, foram utilizados ao longo de sua construção .

Todos esses conceitos oferecem um número significativo de benefícios que não são encontrados em outros modelos, como, por exemplo, organização e estruturação do código fonte que por consequência auxiliam na resolução de problemas complexos.

A linguagem de programação utilizada foi Java. Java é uma linguagem de programação de alto nível orientada a objetos relativamente simples, confiável, portátil, interpretada e de alto desempenho, com suporte a ambientes de tempo real. A tecnologia Java também provê uma plataforma que consiste na integração de *hardware* e *software* necessários para executar programas. A plataforma Java é constituída de dois componentes fundamentais [27]:

a) A Máquina Virtual Java (JVM - *Java Virtual Machine*): é a base da plataforma Java. Foi portada para várias plataformas de *hardware* a fim de se conseguir compatibilidade.

b) A Interface de Programação de Aplicações (API - *Java Application Programming Interface*): é uma grande coleção de componentes de *software* já prontos que provê capacidades extremamente úteis. São agrupadas em bibliotecas, conhecidas como *packages*, ou classes e interfaces.

Como plataforma independente do meio, Java pode ser um pouco mais lenta que códigos nativos. Contudo, vantagens adquiridas a partir da compilação e da tecnologia de máquina virtual trazem desempenho muito perto do esperado em arquiteturas nativas.

Java foi projetada para resolver alguns desafios de desenvolvimento de aplicações em ambientes heterogêneos, principalmente sistemas distribuídos como a *Internet*.

Paralelamente, provê um ambiente seguro com consumo mínimo dos recursos do sistema, implementado alta escalabilidade com a vantagem de portabilidade em qualquer plataforma de hardware ou *software*, inclusive em dispositivos embarcados[27].

O Java oferece uma biblioteca para exibição de gráficos de alta qualidade denominada JFreeChart[28]. Essa biblioteca é 100% Java, facilitando muito o trabalho dos desenvolvedores.

O JFreechart contém uma API bastante consistente e bem documentada, suportando uma grande gama de tipo de gráficos. Possui um *design* bastante flexível

e modularizado, tornando fácil adaptar a sua aplicação de forma independente tanto para o cliente quanto para o servidor.

Ela suporta vários tipos de saída, componentes *Swing* (conjunto de componentes de interface gráfica do usuário - GUI Java) arquivos de imagem como PNG e JPEG, além de formatos de arquivos gráficos vetorizados como o PDF, o EPS e o SVG.

JFreeChart é “open source”, ou, mais especificamente, é *software* livre. Ele é distribuído sob os termos da GNU *Lesser General Public License* (LGPL)[4], que permite o uso de aplicativos proprietários.

5 *Desenvolvimento e Resultados*

5.1 Percurso cognitivo e definição das melhorias

Conforme apresentado no capítulo referente à metodologia, o percurso cognitivo é dividido em duas etapas. A primeira fase, a preparatória, foi definida na seção 4.1. Já a aplicação da segunda fase desse método de avaliação de interfaces gráficas, fase analítica, será descrita nesse capítulo.

Nos comentários que seguem, será apresentada a avaliação de cada tarefa de forma separada. As ações necessárias para completar uma determinada tarefa foram executadas sequencialmente, sendo que as perguntas mostradas na tabela 1 foram consideradas de forma que o avaliador pudesse classificar um possível defeito na interface gráfica. Nessa seção essas perguntas serão referenciadas como critérios, pois durante o método elas são os critérios de avaliação da pessoa que faz a análise.

Cabe adiantar que a discussão da resposta para todas as perguntas de cada ação tornaria o texto muito extenso. Portanto, serão apontados durante a análise, apenas os casos problemáticos ou duvidosos, omitindo os critérios que estão sendo respeitados nessa análise.

Depois de detectar os possíveis problemas da interface através do método do percurso cognitivo, é preciso planejar melhorias para cada tarefa disponibilizada via interface gráfica. Note que a resolução de alguns problemas pode exigir alterações muito drásticas no protótipo, tornando o retrabalho muito grande. Nesse caso serão propostas melhorias que minimizem o problema tornando-o menos perceptivo.

Além disso, não existe uma resposta única e certa para cada problema, visto que esta dependerá da interpretação do analista em detectá-los, e elabo-

rar soluções para minimizá-los.

Nessa seção também serão discutidas as melhorias buscando atender a cada ponto detectado durante a execução do percurso cognitivo. Essas melhorias buscam atender os critérios levantados de forma a não abrir precedentes para outros critérios.

Cada tarefa terá suas ações problemáticas discutidas juntamente com uma proposta de melhoria.

Tarefa 1: Abrir arquivos e selecionar canais

1. Clicar na opção abrir - Nenhum problema detectado
2. Selecionar o arquivo de configuração dos canais que o usuário pretende estudar.

Critério: O elemento de interface revela seu propósito e comportamento?

Na tela de abrir, ilustrada pela Figura 8 na seção 3.1, o usuário precisa saber previamente que o arquivo de configuração do padrão COMTRADE é aquele cuja extensão é “.cfg”. Sem essa informação o usuário não teria informações suficientes para prosseguir na tarefa.

Solução planejada: Para que o usuário saiba que o arquivo de configuração deve ser aquele com a extensão “.cfg”, basta incluí-la na caixa de combinação do campo “Arquivos do tipo”. A Figura 18 mostra o que é uma caixa de combinação. Dessa forma o usuário conseguirá distinguir o tipo de arquivo que deve ser carregado.

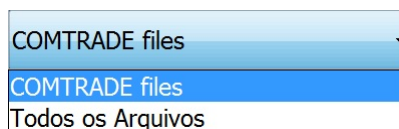


Figura 18: Elemento de interface para a caixa de combinação.

3. Selecionar quais canais analógicos e digitais do arquivo serão carregados.

Critério: Que ações a interface torna disponíveis?

O botão de “OK” da tela de seleção de canais do arquivo (Figura 9) fica habilitado mesmo quando nenhum canal está selecionado, ou seja, a

interface disponibiliza um elemento que não fará o usuário evoluir na tarefa que deseja concluir.

Solução planejada: Será feito um tratamento simples para que o botão “OK” fique desabilitado sempre que não houver canais selecionados nessa tela.

4. Finalizar tarefa.

Critério: Como a interface apresenta o resultado de cada ação?

Após escolher os canais e clicar no botão “OK”, nenhuma mensagem ou alteração significativa na interface é mostrada sinalizando a finalização da tarefa com sucesso. O usuário possivelmente não saberá se as ações realizadas foram feitas de forma correta, caracterizando um ponto do aplicativo que pode ser melhorado.

Solução planejada: A melhoria planejada para essa ação é um pouco mais complexa do que as anteriores. Poderia ser mostrada simplesmente uma mensagem alertando que os canais foram carregados corretamente. Porém, esta seria uma interface a mais que o usuário teria que interagir, o que não é muito interessante. Com isso decidiu-se fazer uma tabela lateral mostrando os canais carregados que será exibida após o clique no botão “OK” da janela de seleção de canais do arquivo. Assim o usuário saberá que concluiu a tarefa de abrir um arquivo e selecionar os canais que deseja trabalhar.

Tarefa 2: Traçar gráficos.

1. Clicar no botão traçar gráficos.

Critério: Onde está o elemento de interface correspondente ao próximo passo?

Os botões que abrem a tela de configuração para exibição de gráficos, janela mostrada na Figura 11, não são muito familiares a outros *softwares* comumente utilizados pelo perfil de usuário definido na seção anterior. Possivelmente o usuário não encontrará os elementos de interface correspondentes a tal ação.

Solução planejada: Aproveitando a melhoria proposta para a ação 4 da tarefa 1, a mesma tabela será utilizada como mais uma forma de chamar a janela de configuração para a exibição de gráficos. Clicando com o botão direito do mouse em algum canal dessa tabela, uma caixa de diálogo se abrirá tendo como uma das opções a chamada para a janela em questão. Pretende-se utilizar tal tabela para chamar as outras ferramentas disponibilizadas pelo WAPS.

2. Escolher quais canais serão traçados, dividindo-os entre as quatro áreas de plotagem.

Critério: O elemento de interface revela seu propósito e comportamento?

Enquanto o usuário está movimentando os canais para uma das quatro áreas, ele não consegue visualizar se as outras áreas representadas pelas quatro abas do painel tabulado superior estão vazias ou não. A Figura 19 mostra o painel tabulado em questão.

Solução planejada: Sempre que alguma área do painel tabulado da tela de configuração para a exibição de gráficos for preenchida com pelo menos um canal, o título dela será colocado em negrito ou itálico (qual se mostrar melhor na interface). Sempre que essa área ficar vazia, o título da aba volta ao normal. Dessa forma o usuário saberá quais áreas então vazias e quais possuem pelo menos um canal.

3. Finalizar tarefa - Nenhum problema detectado.

Tarefa 3: Movimentar cursores.

1. Clicar na barra de opções.

Critério: O elemento de interface revela seu propósito e comportamento?

Critério: O usuário saberá por onde começar? Saberá qual é o próximo passo?

A Figura 13 mostra a tela de exibição dos gráficos. O usuário não sabe que o painel de opções contém os botões que habilitam o cursor. A interface leva o usuário a clicar no cursor antes de habilitá-lo no painel, ou seja, a

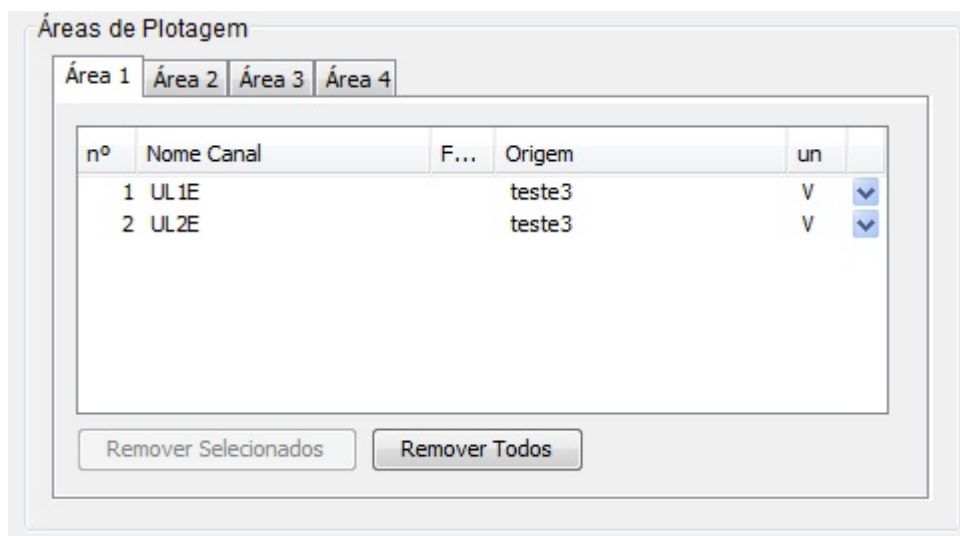


Figura 19: Painel tabulado superior.

interface não revela seu propósito e com isso o usuário não sabe como deve prosseguir.

Solução planejada: Conforme avaliado neste documento, a tela de exibição de gráficos leva o usuário a clicar no cursor para tentar habilitá-lo, com isso, seria interessante implementar essa forma de habilitação para o cursor. Sempre que o usuário passar com o *mouse* por cima do cursor, pode-se alterar o ícone do *mouse* a fim de mostrar para o usuário que existe tal ação implementada.

Para que o usuário saiba o próximo passo, sempre que ele movimentar o cursor, o painel de opções torna-se habilitado, afinal, o cursor perde a utilidade quando o painel de opções está desabilitado, pois a única forma de se visualizar os valores apontados pelos cursores é através das tabelas de tal painel.

2. Habilitar um dos cursores.

Critério: Onde está o elemento de interface correspondente ao próximo passo?

O botão que ativa os cursores não possui nenhuma identificação a respeito de sua funcionalidade, e é a única forma de ativar os cursores.

Solução proposta: O painel de opções ainda está bem enxuto, portanto se forem adicionadas na interface *labels* de identificação em cima dos

botões dos cursores, a mesma não será prejudicada, aumentando a quantidade de informação para o usuário. Assim ele saberá a utilidade dos botões de ativar cursores.

Critério: O resultado apresentado tem correspondência com o objetivo do usuário?

Não tem como desativar os dois cursores, ou seja, retornar ao estado inicial onde ambos estão desabilitados. Quando o botão perde o foco, não dá para saber qual cursor ele habilita/desabilita.

Solução proposta: O botão que ativa os cursores é aquele convencional, que não se mantém pressionado. Trocando o tipo de botão para aqueles que alternam o estado (pressionado e não pressionado), o usuário saberá que o botão que desabilita o cursor que está sendo movimentado é aquele com estado pressionado. Para desabilitar os dois cursores, basta alterar o estado dos dois botões de alternar para não pressionado.

3. Selecionar uma curva na tabela de operações.

Critério: O usuário consegue identificar os elementos de interface?

No funcionamento atual do *software*, o usuário deve selecionar as duas curvas que deseja estudar pelos cursores na tabela de operações, mas, possivelmente, o usuário não irá identificar que ele deve fazer essa seleção, o que mostra um ponto onde a interface não está sendo identificada por aquele que a utiliza.

Solução planejada: Se a interface da tela de exibição de gráficos se comportar de maneira que sempre que um cursor se tornar habilitado, uma curva seja selecionada automaticamente (por definição sempre a primeira curva da primeira área), a atenção do usuário se voltará para a tabela de operações e, dessa forma, acredita-se que ele tem maiores chances de descobrir que a caixa de combinação é responsável pela seleção das curvas.

Critério: Como a interface apresenta o resultado de cada ação?

Os cursores do gráfico não possuem identificação, portanto, não dá para saber qual é o cursor 1 e qual é o cursor 2. Quando se escolhe a curva na caixa de combinação da tabela de operações, não se sabe para qual dos cursores no gráfico cada caixa corresponde.

Solução planejada: Para que seja mais fácil identificar qual caixa de combinação da tabela de operações é responsável por qual cursor, será adicionado um *label* “C1” em cima do cursor 1 e “C2” em cima do cursor 2 do gráfico.

4. Habilitar o outro cursor.

Os problemas que essa ação possui estão descritos no item 2 dessa tarefa.

5. Selecionar outra curva.

Os problemas que essa ação possui estão descritos no item 3 dessa tarefa.

6. Arrastar cursores e comparar curvas.

Critério: O que o usuário vai tentar fazer a cada momento?

A maioria dos programas que apresentam cursores para auxiliar no estudo de uma determinada curva, os mesmos são movimentados pela ação de arraste do mouse.

Solução projetada: Como a maioria dos programas semelhantes ao programa avaliado apresenta a movimentação do cursor por arraste, visando facilitar a aprendizagem do WAPS, será implementado esse tipo de movimentação para os cursores.

Tarefa 4: Exibir e analisar fasores.

1. Alternar para tela de exibição de fasores.

Critério: O usuário consegue identificar os elementos de interface?

A Figura 15 citada na seção 3.2, mostra os gráficos fasoriais. Observe que o usuário não tem como saber de qual área de plotagem pertence às curvas exibidas nos gráficos.

Solução planejada: Propõe-se colocar um *label* em cima dos painéis de plotagem onde estão traçados os gráficos fasoriais indicando a área que os mesmos pertencem.

2. Movimentar cursores - Nenhum problema foi detectado.

3. Comparar curvas

Critério: Como a interface apresenta o resultado de cada ação?

Foram feitos testes em tela com diversas resoluções, e observou-se que em algumas delas, como por exemplo, 1280x768 *pixels* sempre que o usuário movimenta e habilita o painel de opções, as áreas de plotagem das curvas fasoriais ficam com metade de seu conteúdo oculto, conforme pode ser observado na Figura 20.

Solução planejada: É preciso diminuir o tamanho dos painéis de exibição de gráficos fasoriais de forma que seu conteúdo não fique parcialmente fora do campo de visualização do usuário quando o painel de opções está habilitado para as resoluções comumente utilizadas (1366x768, 1280x768, 1280x600, 1024x768 e 800x600 *pixels*).

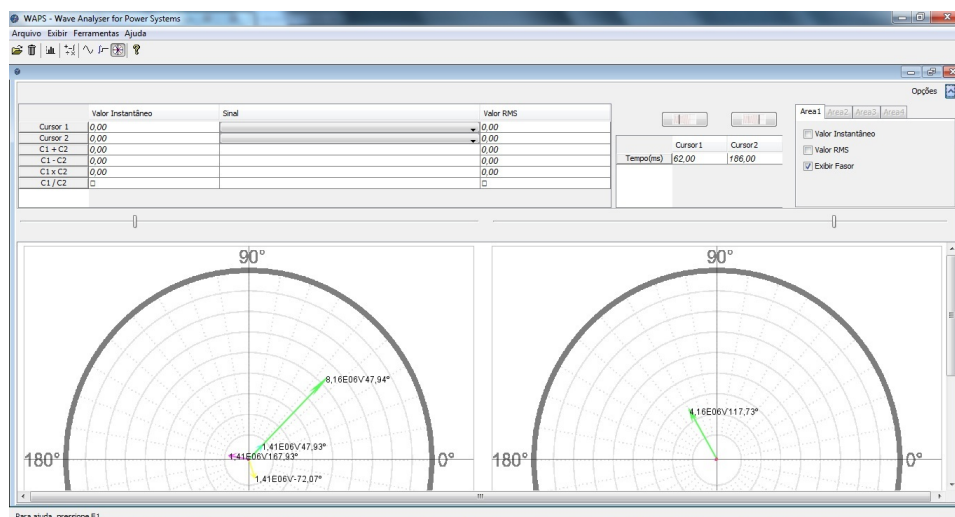


Figura 20: Janela de exibição de fasores com painel de opções habilitado.

Tarefa 5: Ferramenta *Math Builder*

1. Clicar na ferramenta *Math Builder* - Nenhum problema foi detectado.
2. Realizar operação aritmética entre curva e constante.

Critério: Como a interface apresenta o resultado de cada ação?

Após criar um novo canal com o botão “Processar”, Figura 10, a interface não mostra nenhuma mensagem sobre falha ou sucesso na criação, e nem

apresenta qualquer comportamento que indique ao usuário que o canal foi processado, e que um novo canal foi realmente criado. Mais uma vez o aplicativo cria uma situação onde ele não apresenta o resultado da ação que foi tomada pelo usuário.

Solução planejada: Após criar um novo canal com o botão “Processar”, a interface retornará o canal criado dentro do campo de visualização da tabela utilizada para a seleção do primeiro argumento das operações. Assim o usuário saberá que o canal foi criado corretamente. Além disso uma janela com uma mensagem indicando o êxito na criação do canal será exibida para tornar mais evidente o resultado dessa ação.

3. Realizar operação aritmética entre duas curvas.

Critério: Como a interface apresenta o resultado de cada ação?

Mesmo problema citado no item 2 dessa tarefa.

Critério: Que ações a interface torna disponíveis?

Ao clicar em curva, uma tabela para a escolha do segundo argumento passa a ser exibida. Tal tabela pode ser observada na Figura 21. Essa tabela exibe todos os canais carregados, inclusive os canais derivados FFT, que não podem ser utilizados como argumento. Mais uma vez o aplicativo torna disponível um elemento de interface que pode levar o usuário a tomar uma ação errada.

Solução planejada: Para que o usuário fique impossibilitado de selecionar canais derivados como o segundo argumento de uma operação aritmética entre curvas, sempre que ele clicar no botão de opção “curva”, Figura 21, todos os canais derivados FFT presentes na tabela de seleção do segundo argumento serão desabilitados.

4. Aplicar FFT em uma curva.

Critério: Como a interface apresenta o resultado de cada ação?

Quando o usuário clica na operação FFT, o painel de parâmetros passa a ser exibido. No entanto, por outro lado, o painel de operações deixa de exibir a operação de FFT, conforme pode ser observado na Figura 22.

Solução planejada: Será feito um tratamento na barra de rolagem do painel de seleção de operações de forma que a operação FFT fique dentro

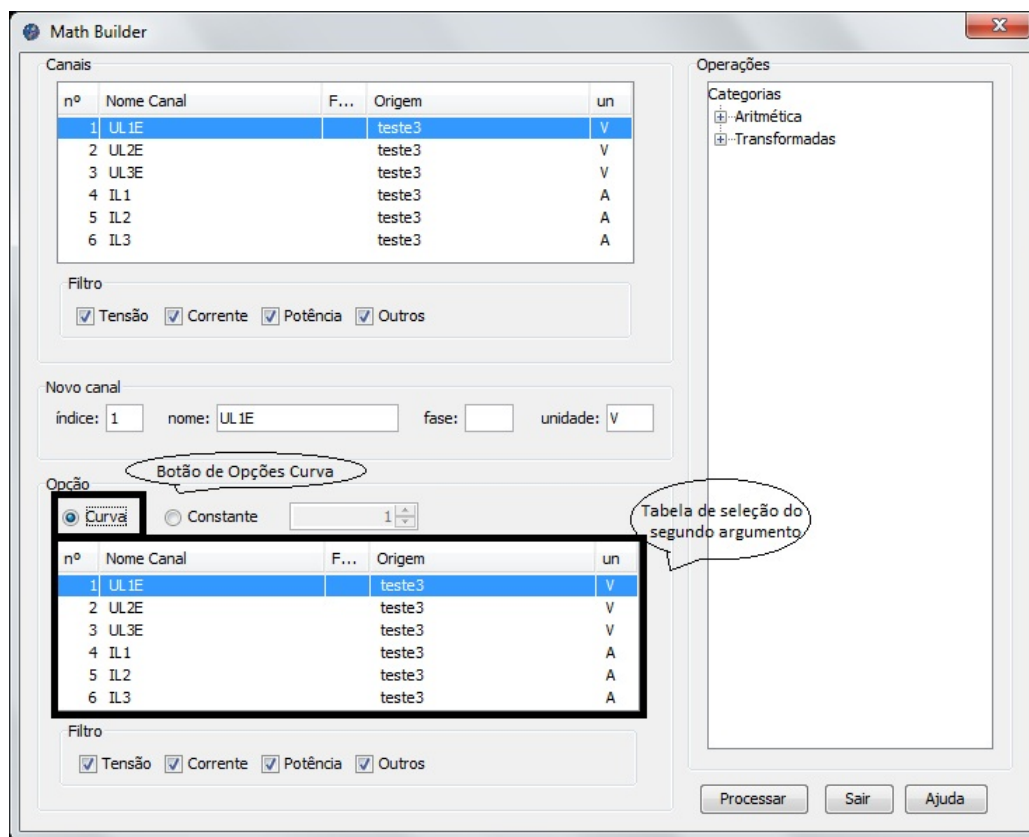


Figura 21: Tabela de seleção do segundo argumento da operação e botão de opções “curva”.

do campo de visualização sempre que o usuário clicar nela. Dessa forma não será preciso fazer nenhuma operação no painel de parâmetros.

5. Aplicar a anti-transformada em uma curva.

Critério: Como a interface apresenta o resultado de cada ação?

Mesmo problema citado no item 2 dessa tarefa.

6. Sair da tela *Math Builder* - Nenhum problema foi detectado.

Tarefa 6: Descartar canais

1. Clicar na opção descartar canais - Nenhum problema foi detectado.
2. Selecionar os canais que serão descartados.

Critério - Que ações a interface torna disponíveis?

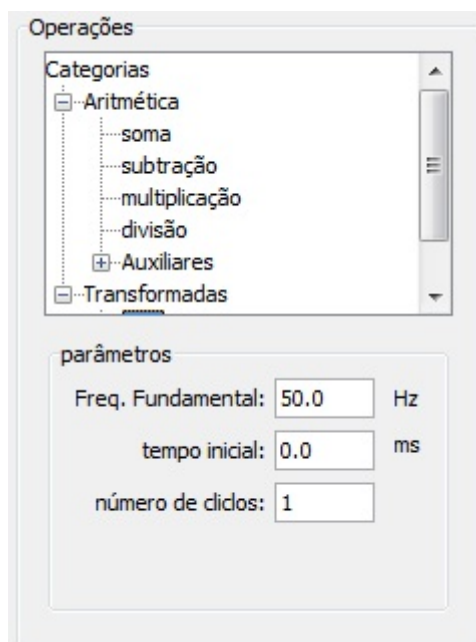


Figura 22: Painel de parâmetros da janela *Math Builder*.

O botão “Nenhum” da tela de descarte de canais na aba de canais digitais, Figura 12, fica habilitado mesmo quando nenhum canal está selecionado. Esse caso pode também ser considerado uma falha de interface gráfica, pois o botão não apresenta nenhuma utilidade podendo, mais uma vez, induzir o usuário a realizar uma ação equivocada.

Solução planejada: Sempre que a aba de canais digitais da tela de descarte for selecionada, analogamente as outras abas, se não houverem canais selecionados, o botão “Nenhum” ficará desabilitado.

3. Sair da tela de descarte de canais - Nenhum problema foi detectado.

5.2 Implementação das melhorias na interface do aplicativo e avaliação dos resultados

Nesta seção será apresentada a implementação das melhorias necessárias em função dos problemas detectados anteriormente. Conforme comentado, o *software* foi implementado utilizando a metodologia de desenvolvimento incremental. Com isso, as melhorias foram aplicadas em cada incremento de forma separada e, ao término de cada incremento, foram incorporadas ao aplicativo final.

Este capítulo apresentará juntamente com as alterações feitas, uma breve discussão dos resultados, buscando avaliar as vantagens e/ou desvantagens que cada melhoria trouxe para a interface gráfica do WAPS.

Dentre as melhorias implementadas temos algumas muito simples, como a adição de uma *label*, outras que parecem simples, porém envolveram uma série de mudanças na estruturação do código, e outras ainda responsáveis por muito alterar a interface do WAPS.

5.2.1 Tela Principal

A Figura 23 mostra a tela de abertura (abrir) com a alteração na caixa de seleção “Arquivos do tipo”. Essa mesma, tela antes das alterações, pode ser observada na Figura 8. Essa melhoria foi classificada como simples, mas, apesar de ser simples, previne que o usuário fique sem saber qual é a próxima ação que deve ser tomada, que nessa ocasião seria selecionar o arquivo “.cfg”.

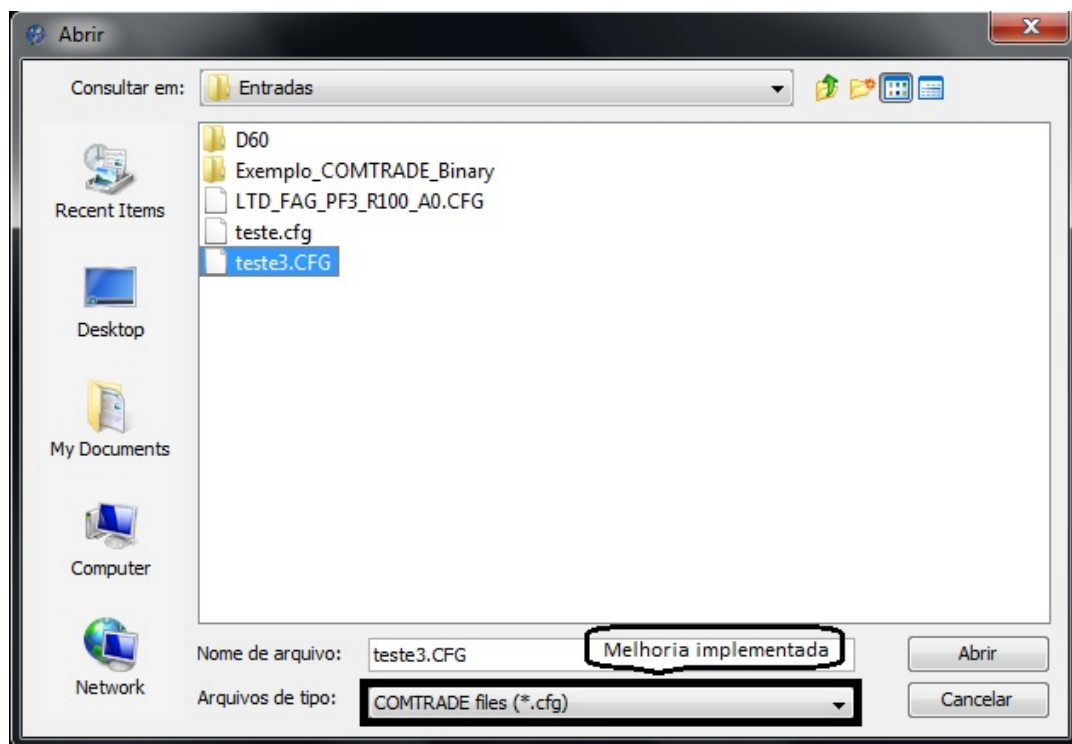


Figura 23: Janela de abrir após a melhoria implementada.

Outra alteração feita na tela principal foi a adição da tabela de canais carregados. Conforme dito anteriormente, essa tabela tem como objetivo mostrar para

o usuário os canais carregados, permitindo que o usuário saiba todos os canais que foram carregados desde que a aplicação foi inicializada.

Esse tipo de tabela possui o nome de árvore. Desta maneira, a mesma será referenciada como árvore de canais carregados. A Figura 24 mostra a sua interface gráfica.

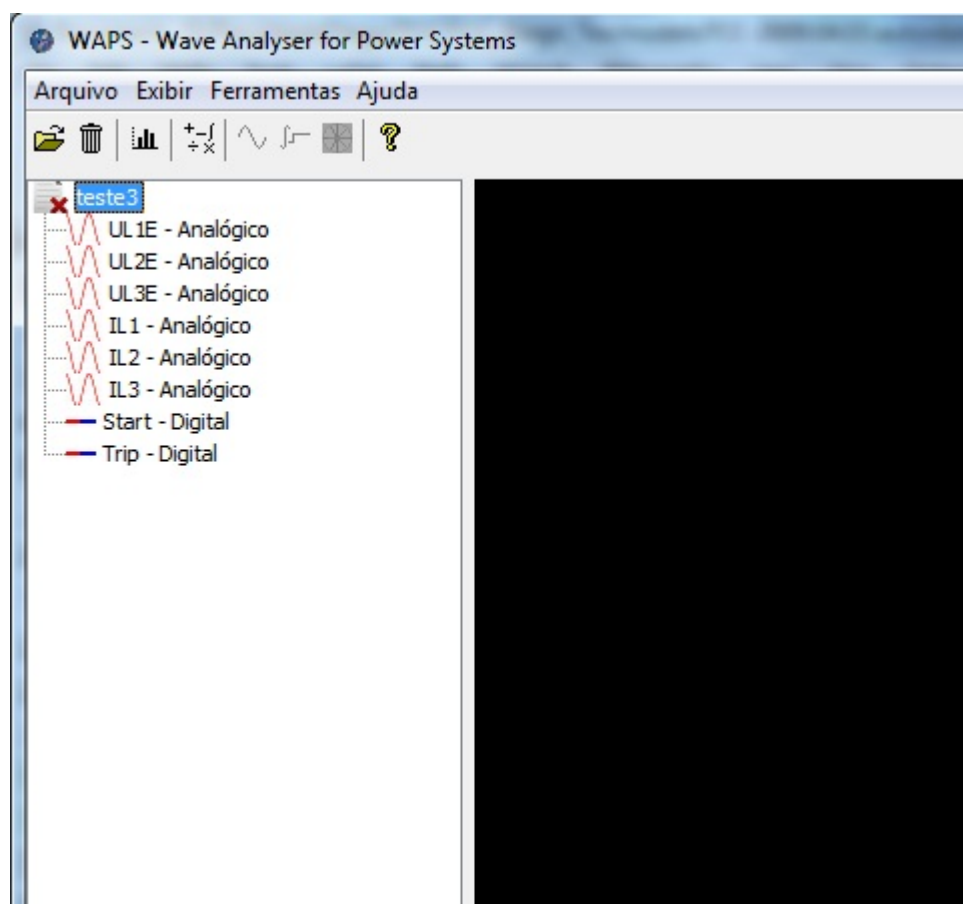


Figura 24: Árvore de canais carregados.

Essa árvore será exibida sempre que o usuário carregar algum canal, e será desabilitada quando nenhum canal for carregado, ou quando todos os canais forem descartados. Na raiz, conforme mostrado na Figura 24, tem-se o arquivo de origem, e nas folhas os canais carregados desse arquivo. A árvore de canais carregados possui as seguintes opções:

Clicando com o botão direito do mouse dentro do painel da árvore de canais carregados, porém sem selecionar nenhuma raiz ou nó, tem-se o menu *pop-up* mostrado na Figura 25a com as seguintes funcionalidades:

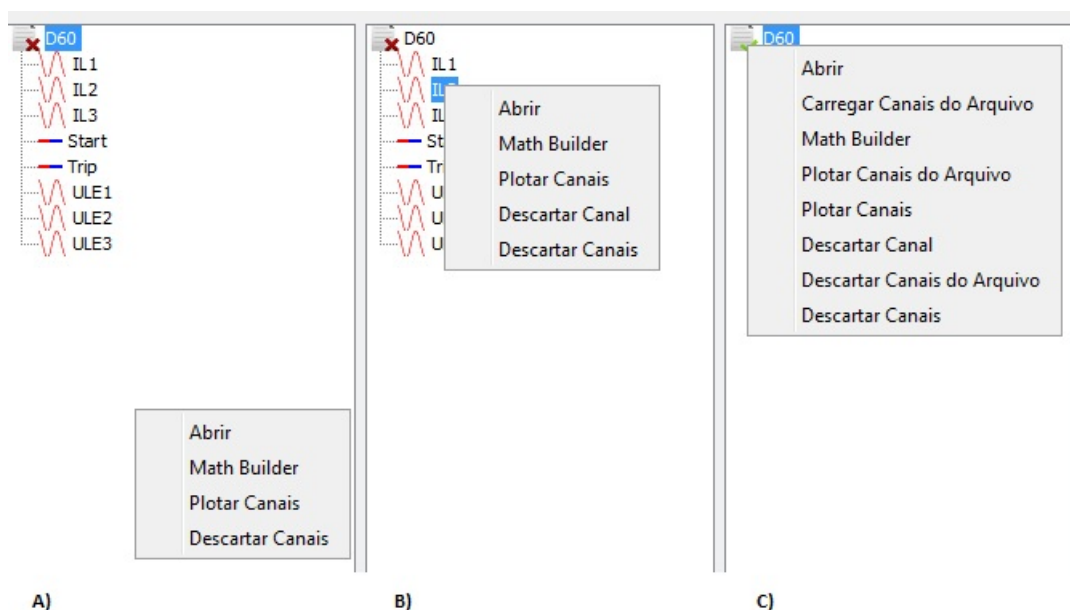


Figura 25: Menu *Pop-up* da árvore de canais carregados.

Abrir - Chama a janela de abrir

Math Builder - Chama a janela *Math Builder*

Plotar canais - Chama a janela de “configuração para a exibição de gráficos”

Descartar canais - Chama a janela de descarte de canais

Clicando com o botão direito do mouse em cima de uma folha, abrirá o menu *pop-up* mostrado na Figura 25b com as seguintes opções:

Descartar canal - Esta opção irá descartar apenas o canal selecionado

Abrir - Chama a janela de abrir

Math Builder - Chama a janela *Math Builder*

Plotar canais - Chama a janela de “configuração para a exibição de gráficos”

Descartar canais - Chama a janela de descarte de canais

Clicando com o botão direito em cima de uma raiz da árvore se abrirá o menu *pop-up* mostrado na Figura 25c com as seguintes funcionalidades:

Carregar canais do arquivo - Abre a janela de seleção de canais referente ao arquivo selecionado.

Plotar canais do arquivo - Abre a janela de “configuração para a exibição de gráficos” permitindo que apenas os canais do arquivo selecionado possam ser plotados.

Descartar canais do arquivo - Abre a janela de descarte dos canais permitindo que apenas os canais do arquivo selecionado possam ser descartados.

Abrir - Chama a janela de abrir

Math Builder - Chama a janela *Math Builder*

Plotar canais - Chama a janela de “configuração para a exibição de gráficos”

Descartar Canais - Chama a janela de descarte de canais

É importante lembrar que essa alteração partiu da motivação de ter alguma reação da interface gráfica após o usuário pressionar o botão de “OK” da janela de seleção de canais. A outra motivação era disponibilizar mais uma forma de chamar a tela de “configuração para a exibição de gráficos”.

Com essa árvore de canais carregados, os dois problemas de interface gráfica em questão foram solucionados e, além disso, aproveitou-se para adicionar as outras opções descritas acima, de forma que nenhuma nova ferramenta precisasse ser implementada, e por outro lado, oferecendo mais possibilidades para o usuário concluir seus objetivos sem tornar seu ambiente de trabalho muito carregado de informações.

5.2.2 Janela de Seleção de Canais

A primeira alteração feita nesse incremento partiu da análise que a interface gráfica não deve disponibilizar ações que façam o usuário agir de forma errada. Nessa tela (figura 9), conforme comentado anteriormente, o usuário tinha a possibilidade de clicar no botão “OK” mesmo que nenhum canal fosse selecionado, o que não traria nenhuma evolução na tarefa a ser concluída. Além disso, fazer com que o botão de “OK” fique desabilitado, enquanto nenhum canal for selecionado, fará o usuário pensar que precisa tomar alguma atitude para torná-lo habilitado, favorecendo a sua aprendizagem.

Após a implementação da melhoria, sempre que o usuário chegar a essa janela,

caso nenhum canal desse arquivo tenha sido carregado anteriormente, o botão de “OK” estará desabilitado. Caso ele selecione uma curva, o botão de “OK” passa a estar habilitado e, sempre que a pessoa que está utilizando o *software* clicar em “Nenhum”, o botão de “OK” torna-se desabilitado novamente.

Outro detalhe importante acrescentado nesse incremento do aplicativo é que sempre que o usuário clicar no botão de “OK”, a árvore de canais carregada se tornará habilitada, pelos motivos já discutidos na descrição da implementação do incremento da tela principal.

5.2.3 Janela de configuração para exibição dos gráficos

No incremento da janela de configuração para exibição de gráficos foi feito apenas uma melhoria, sendo esta considerada simples pelos desenvolvedores, agregando contudo, ganhos para a interface gráfica.

Recordando o problema detectado, caso o usuário pretenda exibir curvas em mais de uma área de plotagem, ele deve separar os canais entre as abas da tela de configuração para exibição de gráficos conforme o desejo de visualização. Após preencher uma área e trocar de aba para preencher outra área, a interface não deixava evidente para o usuário qual aba já foi preenchida, ou seja, em qual área ele colocou pelo menos um canal.

Com isso, sempre que alguma das abas da tela em questão for preenchida, o título da aba ficará em negrito, conforme mostrado na Figura 26. Observe que a área dois tem pelo menos um canal, pois o título dela está em negrito. A aba três, que não possui nenhum canal, está com a fonte do título sem nenhuma formatação especial. Sempre que o usuário transferir pelo menos uma curva para alguma das abas, ela passará a ficar com o título em negrito, e quando ele retirar todas as curvas de uma aba, ela passará a ficar com seu título sem nenhuma formatação especial.

A maior dificuldade para essa melhoria é a diversidade de casos de teste que a tela de configuração para exibição de gráficos trás. Como são muitas as formas de transferência das curvas entre as abas, acabam sendo necessários muitos testes, até que o desenvolvedor tenha segurança para afirmar que a interface apresenta o comportamento esperado para qualquer ação do usuário dentro da janela em

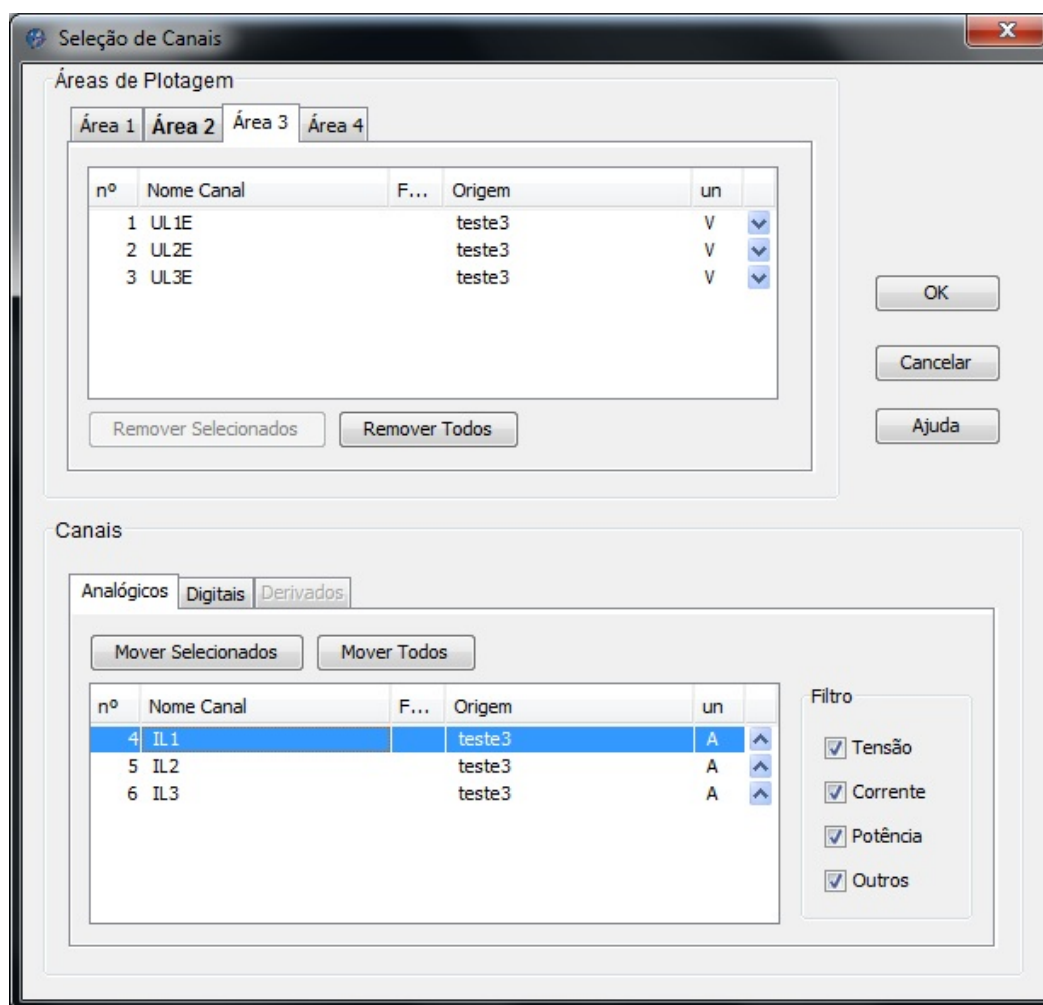


Figura 26: Melhoria implementada na janela de configuração para exibição de gráficos.

questão.

5.2.4 Janela de exibição de gráficos

Essa foi à janela que sofreu a maior quantidade de alterações na interface gráfica, e muitas dessas alterações necessitaram, de certa forma, mudanças complexas no código desse incremento. Isso mostrou que o método do percurso cognitivo deveria ter sido empregado em uma fase menos avançada do projeto, pois evitaria o retrabalho que as melhorias propostas para esse incremento necessitaram.

A primeira melhoria implementada foi um evento de movimentação de mouse, buscando detectar a posição dos cursores no gráfico. Basicamente, tem-se um

ouvinte que fica o tempo todo perguntando se o mouse está passando sobre um dos cursores no gráfico. Quando de fato ele detecta o cursor, o ícone do mouse é modificado para o ícone de redimensionamento, e quando ele sai de cima do cursor, o ícone volta ao normal. É importante ressaltar que para não comprometer o desempenho do aplicativo, esse ouvinte deve ser implementado de maneira que seja o mais eficiente e enxuto quanto possível.

Após esse ouvinte, foi implementado um evento de clique de mouse que só é disparado quando o cursor muda seu ícone para o tipo cursor de redimensionamento, conforme comentado anteriormente. Essa alteração não é muito simples, porém já havia sido implementada na versão antiga do WAPS (versão sem as melhorias). Esse evento de clique descrito ativa o cursor sobre o qual o mouse está. Além de ativar um dos cursores, ele torna habilitado o painel de opções.

Com essas melhorias já é possível atender a diversos tópicos apontados nas seções anteriores como problemas de interface gráfica. Mas ainda foi preciso de mais alterações nesse incremento.

Foram acrescentados os *labels* em cima dos botões de ativar cursores, vide Figura 27, conforme apontado na seção anterior. Essa alteração também é simples, e evita que o usuário chegue a situação de não saber qual o próximo passo a ser tomado para concluir seu objetivo de movimentar os cursores. Os *labels* em cima dos cursores na área de plotagem também foram adicionados, acrescentando ainda as informações que a interface gráfica fornece para o usuário.

É importante lembrar que o excesso de informação em uma interface gráfica pode torná-la muito sobrecarregada de elementos, fazendo com que o usuário não encontre o que precisa. Esses *labels* adicionados foram julgados pelo avaliador como importantes para o usuário, e os locais onde foram colocados não ficaram sobrecarregados de informação, conforme pôde ser visto na Figura 27. Isso trouxe um aspecto bastante positivo para a melhoria implementada.

De acordo com o que foi proposto na seção anterior, o tipo dos botões de ativar cursor passou a ser aqueles de alternar estado. A vantagem dele é justamente mostrar ao usuário seu estado, ou seja, o botão que apresenta o estado pressionado mostra para o usuário que é responsável por ativar/desativar o cursor em destaque na área de plotagem (aquele cuja largura da linha é maior), e por outro lado, aquele que se encontra com o estado não pressionado é responsável pelo cursor

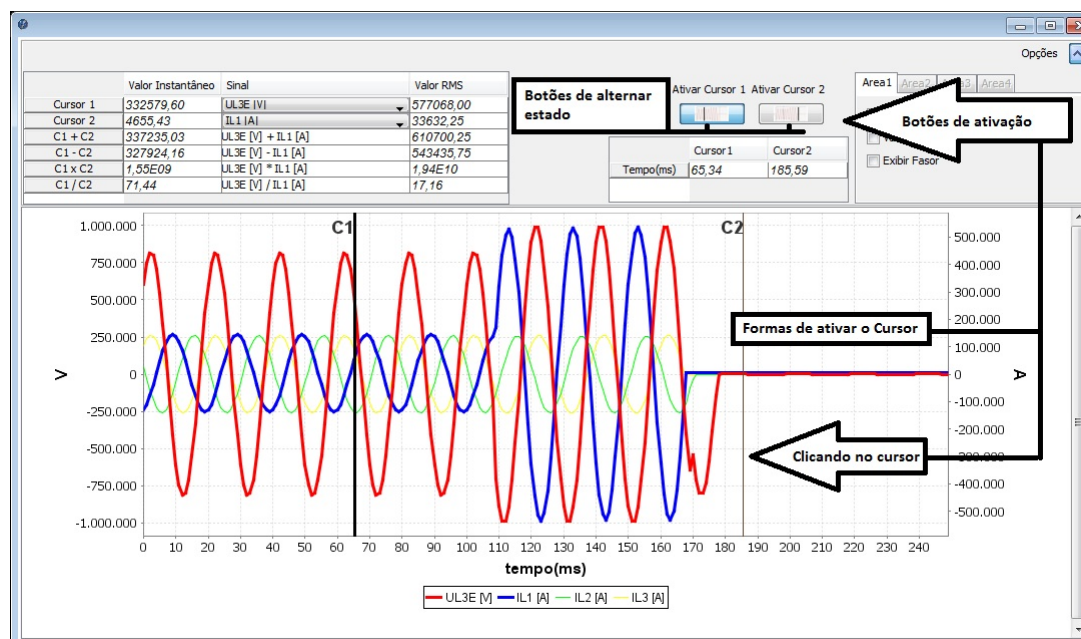


Figura 27: Melhorias implementadas na janela de exibição de gráficos.

que não está em destaque na área de plotagem (aquele cuja largura da linha é menor).

Esse tipo de melhoria é positiva para o *software*, pois ela aumenta a quantidade de informações para o usuário sem aumentar a quantidade de elementos de interface gráfica.

A próxima alteração desenvolvida nesse incremento foi feita a partir da ação de ativar cursores. Com as melhorias implementadas e apresentadas até o momento, existem duas formas de ativar o cursor, que seria clicando em cima dele na área de plotagem, ou clicando em um dos botões de ativar cursor do painel de opções. Sempre que o usuário usar uma dessas duas ações, ela encadeará a seleção automática, na caixa de seleção da tabela de operações, do primeiro canal do tipo analógico disponível na primeira área de plotagem para canais analógicos. Caso esse primeiro canal analógico já esteja selecionado, o segundo será selecionado.

Essa medida na interface gráfica foi tomada também como forma de diminuir as chances do usuário não saber qual é a próxima ação que ele deve tomar para concluir seu objetivo final de movimentar os cursores. Sempre que uma curva é selecionada, a largura do traço que a representa aumenta, chamando a atenção do usuário. Com isso acredita-se que ele irá perceber a necessidade em selecionar

uma curva na caixa de seleção da tabela de operações.

A ultima melhoria implementada nesse incremento é a alteração na forma de arrastar os cursores. Para essa alteração foram levados em consideração a grande quantidade de *softwares* de exibição de gráficos que possuem cursores. A maioria deles apresenta o arraste de cursores através do mecanismo *drag and drop* (arrastar e soltar). Como o foco da análise do percurso cognitivo é criar uma interface que facilite a aprendizagem do usuário, decidiu-se alterar a forma de arraste dos cursores do WAPS para esse mecanismo citado.

Algumas dificuldades foram encontradas no desenvolvimento desse mecanismo, pois a ferramenta de *zoom*, que apresenta o mesmo funcionamento pretendido para os cursores, era acionada durante o arraste dos mesmos. Para tanto foi preciso fazer um tratamento que desabilite o *zoom* quando o clique do mouse sobrepuser a posição de um dos cursores na área de plotagem, sendo habilitado somente após o término da ação de arraste do cursor.

Com isso termina-se a descrição de todas as melhorias pretendidas para esse incremento. De forma geral todas elas trouxeram benefícios para a interface gráfica, mas especialmente para esse incremento, algumas acarretaram em alterações um pouco mais complexas do que o esperado, dificultando um pouco essa fase do desenvolvimento.

5.2.5 Janela *Math Builder*

A maiorias das melhorias implementadas nesse incremento tem como objetivo impedir que o usuário tente realizar alguma operação não permitida. A seguir serão descritas as alterações desenvolvidas para a interface gráfica dessa janela.

O primeiro problema de interface gráfica tratado foi a ação do botão “Processar”. Sempre que o usuário processava alguma operação através desse botão, a interface gráfica não demonstrava nenhuma reação que mostrasse de forma significativa que a criação da nova curva ocorreu conforme o esperado.

A primeira ação tomada foi exibir uma caixa de dialogo informativa apontando o status da operação realizada, ou seja, essa caixa de dialogo vai apresentar uma mensagem informando o nome da curva e se ela foi criada com sucesso ou, em que em caso de erro, ela irá apresentá-lo para o usuário. A Figura 28 mostra

a caixa de diálogo quando a curva é criada com sucesso.

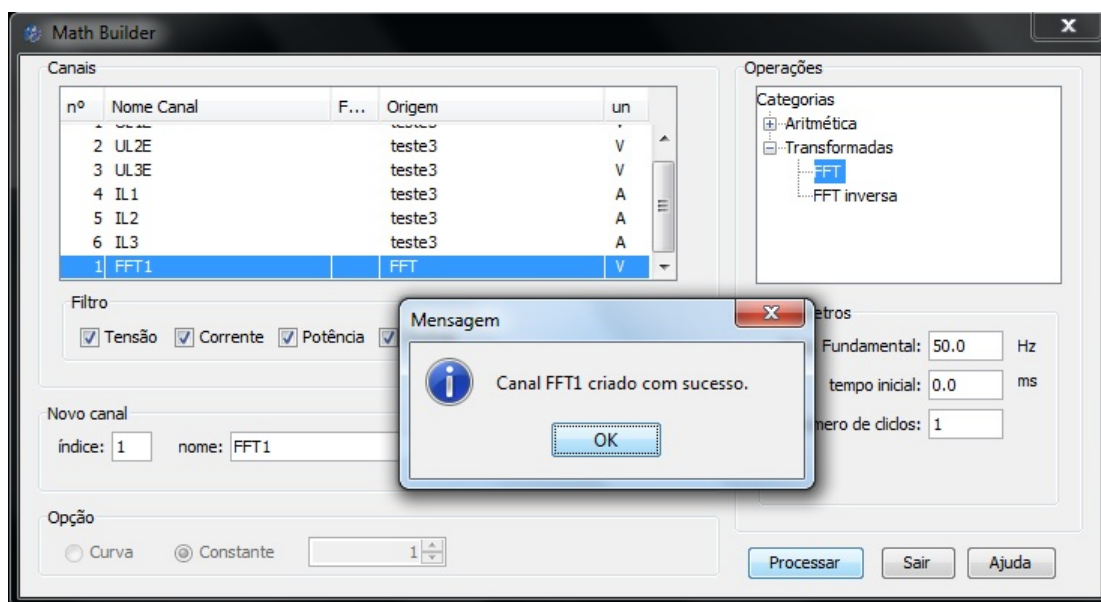


Figura 28: Caixa de diálogo da ferramenta *Math Builder* em caso de operação realizada com sucesso.

Além disso, a tabela de seleção de canais, onde se escolhe o primeiro argumento de uma operação, irá deixar dentro do campo visual a curva criada, mantendo-a selecionada, para chamar a atenção do usuário. Essas ações da interface gráfica acontecem independentemente da operação realizada.

A próxima melhoria tratada foi certificar-se que sempre que o usuário selecionasse uma curva como primeiro argumento, apenas as operações permitidas fossem disponibilizadas, e no caso de operação entre curvas, ou seja, onde o segundo argumento da operação também é uma curva, apenas as séries de canais analógicos e derivados analógicos estariam habilitados na tabela de seleção de canais para o segundo argumento da operação.

Essa medida facilita na aprendizagem do usuário, pois conforme for utilizando o *software* ele irá descobrindo quais tipos curvas podem participar de cada operação disponibilizada pela ferramenta *Math Builder*.

Outro problema de interface tratada acontecia quando o usuário selecionava a operação do tipo FFT no painel de operações. Essa ação habilitava o painel de parâmetros, que por sua vez, toma parte da área antes ocupada pelo painel de operações. O problema é que muitas vezes quando da ocupação dessa área, o painel de parâmetros cobria a operação de FFT que estava selecionada, o que pode

ser uma forma de atrasar a aprendizagem do usuário. Acredita-se que permitindo a operação de FFT dentro do campo visual do usuário fará com que ele associe com mais facilidade o painel de parâmetro a esse tipo de operação, e que isso acelere o processo de aprendizagem do aplicativo.

Para resolver esse problema foi preciso apenas fazer um tratamento na barra de rolagem do painel de operações de forma que ela deixe a operação de FFT dentro do campo visual do usuário quando selecionada. Esse tipo de tratamento é bastante simples, e apesar de não parecer muito significativo para a interface gráfica, torna o *software* mais didático para aqueles que estão começando a utilizá-la.

Com isso, termina-se a descrição das alterações feitas neste módulo. Faltava apresentar apenas o resultado das melhorias do incremento responsável por descartar canais, o que será mostrado a seguir.

5.2.6 Janela de exibição de gráficos fasoriais

Um dos problemas apontados pelo método acontecia durante a ação de movimentar os cursores da janela de exibição de gráficos fasoriais enquanto o painel de opções estava habilitado. Conforme mencionado parte do gráfico ficava fora da área de visualização do usuário causando uma perda nas informações oferecidas pela interface.

A Figura 29 mostra as alterações implementadas na janela em questão. Pode-se observar que os painéis de plotagem tiveram seus tamanhos diminuídos e a legenda foi colocada a esquerda do gráfico. Tais ações permitiram que o gráfico pudesse ser visto por inteiro, sem perder nenhuma informação mesmo quando o painel de opções está habilitado.

Seria preciso fazer um tratamento diferenciado para cada combinação de resolução de tela para que a visualização dos elementos da interface fosse feita sem nenhuma perda de informação para qualquer um dos casos, o que seria inviável. Logo, apenas as resoluções mais usuais foram testadas e devidamente tratadas apresentando a melhoria proposta. Para outros tamanhos de tela, alguns elementos ficam fora do campo de visualização, resultando na não apresentação de todas as informações oferecidas por eles.

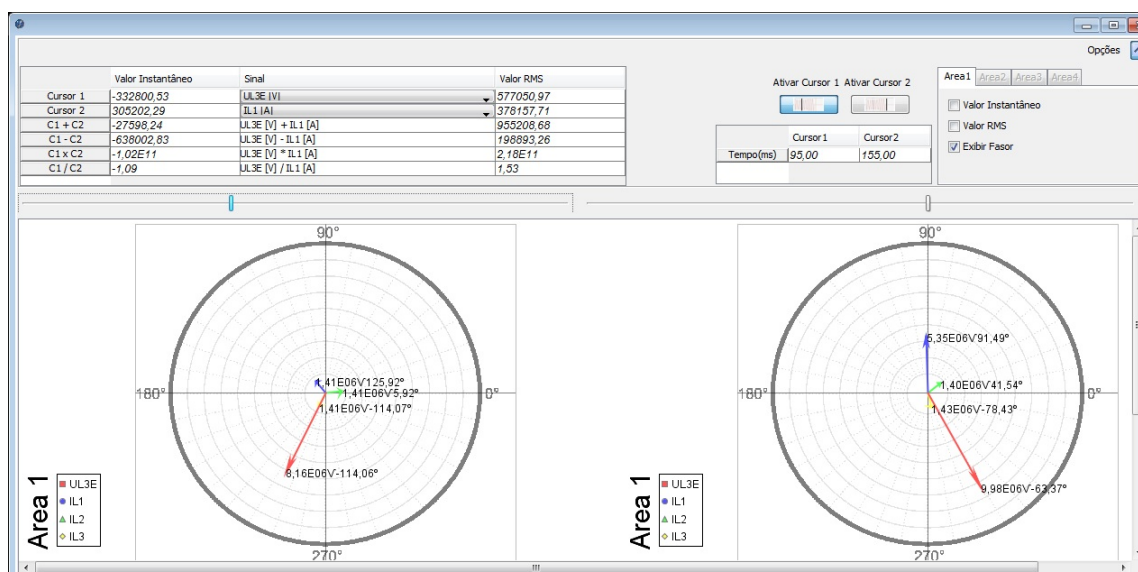


Figura 29: Interface gráfica da Janela de Exibição de Gráficos Fasoriais após a implementação das melhorias.

Outra alteração proposta para esse incremento partiu da necessidade de identificar qual área de plotagem corresponde os fasores exibidos através do gráfico fasorial. A melhoria colocada como solução para esse problema de interface foi justamente a colocação de uma label no painel de plotagem com o nome da área correspondente aos fasor mostrado possibilitando que o usuário consiga identificar o elemento de interface antes não identificado. Esse label foi colocado a esquerda do painel de plotagem justamente para que seja possível de visualizá-lo mesmo quando o painel de opções está habilitado. Essa alteração também pode ser observada na Figura 29.

5.2.7 Janela de descarte de canais

Foi realizado o desenvolvimento de apenas uma melhoria, também com o intuito de evitar que o usuário tome decisões erradas enquanto está utilizando o aplicativo. Essa melhoria faz com que o botão “Nenhum” da aba de canais digitais desse incremento só fique habilitado quando de fato algum canal for selecionado. Afinal, se sua funcionalidade é justamente tirar a seleção de todos os canais, não faz sentido estar habilitado quando nenhum canal se encontra selecionado.

6 *Considerações Finais*

O principal objetivo deste trabalho foi aplicar um método de avaliação de interface gráfica em um *software* para visualização e manipulação de dados oscilográficos fornecidos no padrão IEEE COMTRADE.

Para que fosse possível aplicar esse método, uma fase inicial de análise foi necessária, permitindo que fossem definidos o perfil do usuário, as tarefas que o *software* deveria oferecer juntamente com as ações necessárias para a realização das mesmas, e também quais critérios seriam levados em consideração para o estudo da interface gráfica.

Durante a realização do percurso cognitivo, a maior dificuldade encontrada foi a utilização dos critérios na interface gráfica para detectar os erros. Na execução deste método é preciso que o avaliador se coloque no lugar do usuário, visto que, enquanto tal situação não seja alcançada, a interface gráfica em estudo não irá violar nenhum dos critérios definidos, pois aquele que desenvolve a aplicação não apresenta dúvidas na realização das tarefas, já o usuário pode as possuir.

Outra dificuldade encontrada foi na implementação de algumas melhorias, pois as mesmas necessitaram de mudanças no código fonte não previstas durante as fases de projeto da versão anterior do *software*. Conclui-se com isso que se a aplicação desse método fosse feita em uma fase menos avançada do desenvolvimento, o retrabalho seria menor, e com isso, as implementações das melhorias seriam mais eficientes.

Agora serão explanados de forma crítica os resultados obtidos tanto na fase da aplicação do percurso cognitivo, como na fase do desenvolvimento das melhorias.

A aplicação do percurso cognitivo conforme se esperava, revelou alguns pontos onde o analista avaliou possíveis defeitos de interface gráfica. Esses defeitos partiram de diferentes critérios, o que confirmou a necessidade de uma análise

bem diversificada. Além disso, a quantidade de apontamentos que essa avaliação gerou mostra a real necessidade em se aplicar o método, pois tais pontos passaram despercebidos na criação da primeira versão do módulo.

Outro fator importante de ser colocado sobre a aplicação do método é a praticidade. Esse método mostrou, conforme o esperado, que realmente não necessita de muitos recursos, tanto materiais como humanos, e também de ser bastante prático. Acredita-se que com o aperfeiçoamento dos analistas, a utilização de tal método possa significar muito pouco em relação ao tempo total do projeto, e os resultados gerados possam ser positivos.

Após a utilização do método, veio a fase de implementação das melhorias. Ao longo dessa fase, foi possível observar que algumas modificações apontadas eram bastante simples de serem implementadas, e o resultado final, mesmo não alterando de forma significativa a interface, agia de forma a prevenir possíveis erros que o usuário pudesse cometer enquanto executasse qualquer uma das tarefas estudadas. Outras modificações com essa mesma característica de não alterar de forma significativa a interface, não agiram de forma a prevenir erros, mas sim, como auxiliares para a aprendizagem do usuário.

Como o principal objetivo do método era moldar a interface de maneira que o usuário pudesse aprender a utilizar o *software* de forma rápida e exploratória, os dois pontos levantados anteriormente foram essenciais para garantir o cumprimento desse objetivo levantado.

Outras alterações foram um pouco mais complexas e agiram de forma a aumentar as possibilidades para o usuário. Se o usuário na interface antiga tinha apenas uma forma de avançar na tarefa que estava realizando, após uma alteração desse tipo, passou a ter duas ou mais possibilidades, diminuindo as chances de ficar “preso” em uma determinada ação da tarefa em questão.

É importante levantar alguns pontos que podem complicar ou tornar a aplicação do método do percurso cognitivo, e a implementação das melhorias, ruim para o projeto. É preciso ter muito cuidado com as melhorias implementadas, pois da mesma forma que durante essa fase de desenvolvimento muitos erros antes não encontrados são detectados e corrigidos, outros podem ser gerados durante as implementações das melhorias, o que mostra a necessidade de uma bateria de testes variada e intensa.

Outra questão levantada foi na fase da definição das melhorias. Nesta, apesar de se ter como foco a aprendizagem exploratória do usuário, outros fatores apontados nesse documento, sem o devido cuidado, podem ser prejudiciais à interface gráfica. Como por exemplo, têm-se a eficiência e a confiabilidade, entre outros, que devem ser levados em consideração para que a ação de correção tomada não esteja apenas trocando o tipo de erro encontrado.

De forma geral, cabe colocar que a relação ganhos/retrabalho da interface gráfica foi satisfatória, positiva, e acredita-se que esse tipo de estudo é viável. Conforme comentado, tal estudo poderia ser mais eficiente se aplicado em uma fase anterior ao estado que o aplicativo WAPS se encontrava.

Informações complementares a respeito do aplicativo WAPS podem ser encontradas no anexo deste documento, onde se encontram duas das publicações sobre a ferramenta desenvolvida.

Como sugestão para trabalhos futuros, pode-se citar, por exemplo, a aplicação de outros métodos de avaliação de interface gráfica nesse aplicativo, verificando quais melhorias podem ser apontadas, fazendo-se um estudo comparativo entre os métodos. Outra sugestão seria aplicar o método do percurso cognitivo em outra aplicação, porém, em uma fase do desenvolvimento um pouco anterior a fase aplicada no WAPS, para que seja possível ter uma ideia da quantidade de erros encontrados em relação à necessidade de retrabalho alocado pelas melhorias definidas.

Referências

- [1] SANTOS, A. Q. Waps: Software para visualização e análise de faltas a partir de arquivos do padrão iee comtrade. São Carlos SP, 2011.
- [2] DUGAN, R. C.; MCGRANAGHAN, M. F.; BEATY, H. W. Electrical power systems quality. MCGRAW-HILL, 1996.
- [3] ROCHA, M. C.; BARANAUKAS, M. C. C. *Design e Avaliação de Interfaces Humano-Computador*. NIED/Unicamp: [s.n.], 2003.
- [4] FSF. Gnu lesser general public license. 2007. Disponível em: <<http://www.gnu.org/licenses/lgpl.html>>. Acesso em: 2012.
- [5] WHARTON, C. *The cognitive walkthrough method: a practitioner's guide: Usability inspection methods*. Nova Yorque: Wiley, 1990. 105-140 p.
- [6] NIELSEN, J.; MACK, R. L. *Usability Inspection Methods*. Nova Yorque: Wiley, 1994.
- [7] NIELSEN, J. Usability engineering. Nova Jersey, 1993.
- [8] DALTRINI, J. O. F. de Carvalho e B. M. *Interfaces de sistemas para computadores voltadas para o usuário*. [S.l.: s.n.], 1987.
- [9] MARCUS, A. *Computer Human-Interface: Principles of effective visual communications for graphical user interface design*. São Francisco-CA: Morgan Kaufmann, 1995. 425-441 p.
- [10] MEC. Diretrizes curriculares de cursos da área de computação e informática. p. 12, 2012. Disponível em: <<http://www.inf.ufrgs.br/ecp/docs/diretriz.pdf>>. Acesso em: 2012.
- [11] SNHEIDERMAN, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 2. ed. Massachusetts: Addison-Wesley Publishing Company, 1992. 573 p.
- [12] PRESSMAN, R. S. *Engenharia de Software*. São Paulo: Makron Books, 1995.
- [13] BARFIELD, L. *The user interface: Concepts and Design*. Workinghan: Addison-Wesley Publishing Company, 1993.
- [14] CARROL, J. M.; ROSSON, M. B. *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction: The paradox of the active user*. Cambridge: Bradford Books/MIT Press, 1987.

- [15] LEWIS, C. et al. *Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces*. University of Colorado: Institute of Cognitive Science, 1990.
- [16] RIENMAN, J.; FRANZKE, M.; REDMILES, D. *Usability Evaluation with the Cognitive Walkthrough*. [S.l.]: ACM CHI '95 Proceedings, 1995.
- [17] ABOWD, G. *Performing a Cognitive Walkthrough*. Georgia Institute of Technology: College of Computing, 1995.
- [18] POLSON, P.; LEWIS, C. *Theory-Based Design for Easily Learned Interfaces*. [S.l.]: HumanComputer Interaction, 1990.
- [19] PSRC. Ieee standard common format for transient data exchange (comtrade) for power systems. IEEE Press, 1999.
- [20] LENZ, E. P.; ROLIM, G. J. Aplicação de oscilografias e sequência de eventos para avaliação do desempenho de reles de proteção de geradores. *Revista IEEE, America Latina*, n. 7, 2010.
- [21] ELECTROTEK, C. Top, the output processor: Software for visualizing monitoring data and simulation results. 2011. Disponível em: <<http://www.pqsoft.com/top/>>. Acesso em: 2011.
- [22] SIEMENS. Siga 4 demo en. 2012. Disponível em: <[http://www.energy.siemens.com/ecc_pool/SIPROTEC Programs/ea58a407-4b38-4f3d-b75e-9ee54496aa17/SIGRA4 DEMO en.zip](http://www.energy.siemens.com/ecc_pool/SIPROTEC_Programs/ea58a407-4b38-4f3d-b75e-9ee54496aa17/SIGRA4_DEMO_en.zip)>. Acesso em: 2012.
- [23] DAM, Q. B. *.NET RelayerTM Software Quick Start Guide: IEEE COM-TRADE Viewer and Software for Protective Relay Simulation and Experiments*. [S.l.: s.n.], 2009.
- [24] SINAPE, C. Sistema integrado de apoio a análise de perturbações. 2012. Disponível em: <<http://www.sinape.cepel.br/arquivos/SlideShowDoSinape1.pps>>. Acesso em: 2012.
- [25] GRAINGER, J.; STEVENSON, W. *Power System Analysis*. [S.l.]: McGraw-Hill, 1994.
- [26] GRADY, B. *OBJECT-ORIENTED, Analysis and Design*. 2. ed. [S.l.]: Addison-Wesley, 1993.
- [27] DEITEL, M. H.; DEITEL, J. P. *JAVA, How to Program*. 4. ed. [S.l.: s.n.], 2002.
- [28] GILBERT, D. *The JFreeChart Class Library - Developer Guide*. [S.l.]: Object Refinery Limited, 2007.

Anexo

**Modelo de Orientação a Objetos Aplicado ao Projeto e
Desenvolvimento da Ferramenta Computacional WAPS
para Manipulação e Visualização de Arquivos
Oscilográficos**

Modelo de Orientação a Objetos Aplicado ao Projeto e Desenvolvimento da Ferramenta Computacional WAPS para Manipulação e Visualização de Arquivos Oscilográficos

A. Q. Santos¹, D. C. Grillo¹, U. C. Netto¹, D. Barbosa², D. V. Coury¹ e M. Oleskovicz¹

Resumo—Embora o modelo orientado a objetos esteja bastante consolidado no ambiente de desenvolvimento de *softwares*, existem grandes sistemas legados no setor elétrico atualmente, prevalecendo a arquitetura procedural de desenvolvimento. Este trabalho faz uso dos conceitos intrínsecos ao paradigma de orientação a objetos para a construção de uma ferramenta em código livre denominada WAPS (*Wave Analyser for Power System*) voltada ao estudo e análise de perturbações e transitórios de redes de energia elétrica. Procurou-se analisar e projetar os componentes de *software* de forma a se obter um sistema modular e flexível que se adapte as necessidades do usuário. Como resultado, obteve-se um produto com interface intuitiva e capaz de comportar diversos formatos de entrada. A ferramenta permite, ainda, o trabalho simultâneo entre diversos arquivos oscilográficos com representação distinta para cada tipo de canal. Varias formas de manipulações algébricas sobre sinais estão disponíveis possibilitando a criação de novos canais além de permitir opções de visualização usuais como a representação fasorial ou RMS provendo maior legibilidade das informações.

Index Terms—WAPS, COMTRADE, Orientação a Objetos, Oscilografia, Proteção, Sistemas Elétricos de Potência.

I. INTRODUÇÃO

O aumento do uso de tecnologia digital em dispositivos de proteção, medição e controle, principalmente em subestações de energia elétrica, resultou no acúmulo de grandes quantidades de registros digitais nas mais diversas estruturas, muitas das quais, em formatos proprietários sem qualquer tipo de padronização [1].

Desta forma, a integração entre os diferentes processos de análise e de estudos, nos vários setores da indústria elétrica, é complexa e necessita do conhecimento de múltiplas ferramentas computacionais. Assim, com a necessidade do intercâmbio de informações para melhorar e automatizar procedimentos, surge a conveniência em se adotar um formato unificado e padronizado para registros digitais [2].

Nesse sentido, o *Institute of Electrical and Electronics Engineers* (IEEE) define um formato comum para arquivos

de dados digitais e meio de troca para intercâmbio de vários tipos de faltas, testes e simulações, sendo capaz de representar formas de ondas geradas a partir de registradores de faltas, relés de proteção, medidores de qualidade de energia e dispositivos similares. Tal padrão é conhecido como COMTRADE, cujo acrônimo significa *COMmon format for TRANsient Data Exchange*, e é amplamente adotado tanto pela indústria quanto pela academia.

É possível encontrar na literatura diversos aplicativos, em sua maioria comerciais, e por isso, não acessíveis a todos os profissionais da área, capazes de realizar manipulações ou exibir informações de arquivos oscilográficos [3]–[5]. Dentre os mais utilizados, pode-se citar o SINAPE (Sistema Integrado de apoio à Análise de Perturbações) desenvolvido pelo Centro de Pesquisas de Energia Elétrica da Eletrobrás (CEPEL). Outro aplicativo é o TOP (*The Output Processor*), desenvolvido pela Electrotek Concepts, permite a visualizações dos dados de diversos formatos, entre eles o COMTRADE. O *software* SIEMENS SIGRA é uma ferramenta que reconhece apenas arquivos no formato COMTRADE e está disponível apenas para o sistema operacional Windows, sendo que sua licença não é gratuita. Já o .NET Relayer é um programa simples, anteriormente denominado *Waveform Analyser*, é um visualizador de arquivos COMTRADE cujos cálculos matemáticos precisam ser programados antes de serem exibidos, o que pode ser uma vantagem para cálculos específicos, porém um problema no caso de manipulações muito usuais.

Como no passado, a prevalência do paradigma de programação procedural norteou a grande gama de aplicativos encontrados hoje no setor elétrico, uma grande dificuldade é encontrada atualmente na integração de novos componentes de *software* aos sistemas legados anteriores e grandes investimentos devem ser direcionados para a manutenção e atualização destes sistemas [6].

Neste contexto, este trabalho tem por objetivo aplicar a Análise e Projeto Orientado à Objeto (APOO) para a construção da ferramenta computacional WAPS (*Wave Analyser for Power System*), desenvolvida com todas as suas características e funcionalidades disponíveis em diversas plataformas, sendo capaz de tornar os dados referentes aos fenômenos ligados à geração, transmissão e distribuição de energia elétrica manipuláveis computacionalmente.

¹Universidade de São Paulo (USP) - Escola de Engenharia de São Carlos (EESC) - Departamento de Engenharia Elétrica (SEL), Av. Trabalhador Sancarlense, 400, São Carlos, São Paulo, Brasil. e-mail: athila.santos@usp.br, diego.grillo@usp.br, uchenin@usp.br, coury@sc.usp.br, olesk@sc.usp.br.

²Universidade Salvador (UNIFACS) - Mestrado em Energia - Rua Ponciano de Oliveira, 126, 2 andar, Rio Vermelho, Salvador, Bahia, Brazil. e-mail: daniel.barbosa@pro.unifacs.br.

O *software* foi desenvolvido em módulos interconectados com uma interface de comunicação bem definida a fim de se prover a modularidade necessária para o aperfeiçoamento futuro da ferramenta como a adição de novos formatos de entrada ou a construção novas manipulações matemáticas. O padrão IEEE COMTRADE foi escolhido como o formato inicial para a manipulação de dados dentre os padrões de registros oscilográficos devido ao seu grande uso tanto no meio acadêmico, quanto no meio industrial. Porém, novos módulos de reconhecimentos de formatos podem ser desenvolvidos e integrados junto à ferramenta.

É factível destacar ainda, que o sistema proposto foi elaborado para atender a maioria das necessidades dos usuários e as diretivas de implementação de um aplicativo, tais como: funcionalidade, velocidade, eficiência, confiabilidade, segurança, consistência, dentre outros.

Embora a maioria das funcionalidades implementadas existam nos *softwares* existentes no mercado, o conjunto proposto não é encontrado em um único aplicativo, viabilizando a construção de uma ferramenta modular, de código aberto e livre (*Open Source*), e que pudesse ser utilizada em quaisquer sistemas operacionais, se adaptando aos diversos tipos de canais existentes, tanto na norma COMTRADE, quanto aos novos canais criados dentro de seu domínio e que possuam representações distintas com a aplicação de manipulações algébricas e transformadas de base. Também permite a manipulação de diversos arquivos e formatos, o emprego de cursores e gráficos simultâneos, além de possibilitar o uso de diversas janelas em uma mesma área de trabalho (MDI, do inglês *Multiple Document Interfaces*).

II. PARADIGMAS DE PROGRAMAÇÃO

Um estilo de programação é um modo de organizar programas com base em um modelo conceitual de programação e uma linguagem apropriada para escrevê-los de forma clara [7]. Existem cinco tipos principais de paradigmas conforme descrito na tabela 1.

Tabela I: Paradigmas de Programação

Paradigmas	Exemplificação
Orientada a Procedimentos	Estrutural
Orientada a Objetos	Classes e Objetos
Orientada a Lógica	Objetos frequentemente expressados em predicados
Orientada a Regras	if-then-regras
Orientada a Restrições	Relações

Fonte: [7]

Não existe um estilo de programação que seja melhor que todos os outros, mas apenas situações em que determinados paradigmas são mais apropriados. Por exemplo, programação orientada a regras é mais eficiente para projetos envolvendo bases de conhecimento, enquanto programação estrutural é mais apropriada para problemas que envolvam o uso intensivo das operações computacionais.

A. Orientação a objetos

O termo orientação a objetos é uma quebra de paradigma do desenvolvimento tradicional de *software*, em que estruturas de

dados e rotinas são desenvolvidas de forma apenas fracamente acopladas. É um conceito que pressupõe uma organização em termos de coleção de objetos discretos incorporando estrutura e comportamento próprios. Requer um modo de pensamento diferenciado para decomposição do problema e possui uma arquitetura bastante diferenciada dos métodos de projeto estruturados [8].

A abordagem de orientação a objetos favorece a aplicação de diversos conceitos considerados fundamentais para o desenvolvimento de bons programas. Dentre os conceitos principais envolvidos, pode-se citar:

1) Abstração de dados

Considera-se abstração como o ato de focalizar os aspectos essenciais inerentes a uma entidade e ignorar propriedades específicas, ou seja, concentrar-se na definição de um objeto antes de decidir como será implementado. O uso de abstração preserva a liberdade para tomar decisões de desenvolvimento ou de implementação apenas quando há um melhor entendimento do problema a ser resolvido. Toda variável de uma linguagem de programação é uma abstração, pois representa virtualmente determinado objeto que existe no mundo real.

2) Encapsulamento

O conceito de encapsulamento permite que mudanças em programas sejam mais confiáveis através da restrição de informações. A estrutura de um objeto é escondida assim como a implementação de seus métodos. Desta forma, o encapsulamento separa os aspectos externos de um objeto, os quais são acessíveis a outros objetos, dos detalhes internos de implementação. O uso de encapsulamento permite a modularização do código, evitando que pequenas mudanças tenham repercussões em toda estrutura do código, e melhor reaproveitamento em outras aplicações.

3) Polimorfismo

Determina o comportamento de uma operação podendo ser implementada por mais de um método em uma mesma classe ou ter comportamento distinto em classes distintas. A linguagem de programação deve ser capaz de selecionar o método correto a partir do nome da operação, classe do objeto e argumentos. Desta forma, novas classes podem ser adicionadas sem necessidade de modificação do código já existente devido a mudança em seu escopo.

4) Modularidade

O ato de particionar um programa em componentes individuais reduz significativamente sua complexidade criando um número definido de interfaces. Organizar essas estruturas em módulos produzindo uma arquitetura física ajuda o gerenciamento da complexidade principalmente em grandes aplicações em que se podem ter centenas de classes. Como geralmente os módulos são unidades elementares e indivisíveis de um *software*, estes podem ser reutilizados em outros aplicativos.

5) Herança

Técnicas de orientação a objetos promovem compartilhamento em diversos níveis distintos. A Herança de

estrutura de dados permite que estruturas comuns sejam compartilhadas entre diversas classes derivadas similares, sem redundância. Ainda mais importante que a economia de código é a clareza conceitual do relacionamento e interconexão entre as estruturas de dados, o que reduz o número de casos distintos que devem ser entendidos e analisados.

6) Interface

Uma interface modela um comportamento esperado. Ao se modelar um sistema, pode-se pensar apenas nas interfaces de seus objetos, ou seja, em suas funções e relacionamentos. Cria-se então, uma camada extra de abstração. O uso de interfaces sugere a construção de uma planta do sistema de uma forma consistente e clara facilitando o desenvolvimento posterior dos códigos que seguirão os moldes já existentes.

III. O MODELO ORIENTADO A OBJETOS

Um modelo orientado a objetos busca capturar a estrutura estática de um sistema mostrando os objetos existentes, seus relacionamentos, atributos e operações que caracterizam cada classe. É através do uso deste modelo que se enfatiza o desenvolvimento em termos de objetos ao invés de mecanismos tradicionais de desenvolvimento baseado em funcionalidades, permitindo uma representação mais próxima do mundo real [8].

Objetos são abstrações com limites e significados bem definidos para uma aplicação. Existem dois propósitos em sua utilização: promover o entendimento do mundo real e suportar uma base prática para uma implementação computacional. A decomposição de um problema em objetos não é única; depende do julgamento do projetista e da natureza do problema [9]. Para construir um objeto projeta-se uma classe que irá agrupá-los conforme suas similaridades, relacionamentos e semântica. Classes são definidas a partir de propriedades, referentes aos atributos de um objeto, e comportamentos, que estão ligados a operações e métodos suportados por um objeto.

A. Análise e Projeto Orientado a Objetos

Para a obtenção de melhores soluções quando do desenvolvimento de um *software*, deve-se seguir um processo detalhado para a análise dos requisitos do sistema que se pretende desenvolver. Deve-se fazer, ainda, uma análise do domínio buscando identificar os principais conceitos relacionados ao aplicativo. Posteriormente, faz-se necessário um maior detalhamento dos requisitos e conceitos até então estabelecidos e dos relacionamentos existentes. Durante a implementação do sistema, busca-se atender os requisitos necessários para o desenvolvimento do *software*. Por fim, a fase de verificação e validação é executada para assegurar o atendimento dos requisitos.

Uma das ferramentas utilizadas em projetos de *software* é a UML (*Unified Modeling Language*) que é uma linguagem gráfica que permite a projetistas e desenvolvedores representarem seus projetos orientados a objetos utilizando uma notação comum [10]. Dos diagramas disponíveis na UML aponta-se os diagramas de *Classe*, de *Casos de Uso* e o diagrama de *Sequência* como os mais utilizados.

IV. REQUISITOS DO SISTEMA

A. Requisitos Funcionais

1) *Reconhecimento de diversos formatos de entrada*: O *Softwares* deverá reconhecer os principais formatos de arquivo usados em sua área de atuação. Também deve ser capaz de se adaptar a possíveis atualizações de formatos bem como o surgimento de novos padrões. Uma arquitetura modular facilita a incorporação de diferentes funcionalidades mesmo após o lançamento do aplicativo. Porém, inicialmente apenas o formato IEEE COMTRADE deverá estar disponível como reconhecimento de formato de entrada. O sistema deverá reconhecer tanto arquivos no formato ASCII como no formato de compactação binária.

2) *MDI*: Interface de documentos múltiplos ou MDI (do inglês, *Multiple Document Interfaces*) é um método de organização de aplicações gráficas em painéis que residem em uma janela principal. O aplicativo deverá permitir a utilização de vários ambientes de trabalho na mesma seção conforme os preceitos de MDI.

3) *Operações matemáticas*: Operações matemáticas auxiliam na manipulação dos sinais permitindo aos usuários realizarem as modificações necessárias para a análise de interesse. Desta forma, um módulo do sistema denominado *Math Builder* deverá ser construído para prover manipulações de canais através da criação de novos objetos gerados a partir das operações matemáticas disponibilizadas. Tais objetos herdam as características do canal ao qual foi derivado podendo ser utilizado em novas operações ou na visualização de novos gráficos. Apenas operações sobre canais analógicos e derivados serão permitidos. O escopo das operações é melhor detalhado nos itens a seguir.

- *Operações Aritméticas*: Operações Aritméticas são as operações mais básicas e só podem ser realizadas sobre canais analógicos. Dentre as operações disponíveis estão:
 - soma
 - subtração
 - multiplicação
 - divisão
- *Operações Auxiliares*: As operações auxiliares dão recursos adicionais na manipulação dos sinais e inicialmente apenas duas operações estão disponíveis:
 - negativo($-x$)
 - inverso($\frac{1}{x}$)
- *Operações Transformadas*: Operações de transformadas utilizam um novo domínio de referência diferente do tempo. Este domínio, denominado *Domínio da Frequência*, representa seus sinais em componentes espectrais e por isso existem algumas restrições de operações conforme o tipo de canal. A *FFT* (do inglês, *Fast Fourier Transform*) transforma sinais analógicos discretos no tempo em sinais espectrais no domínio da frequência respectivamente e sua inversa é realizada a partir do algoritmo *IFT* (do inglês, *Inverse Fourier Transform*).
 - *FFT*
 - *FFT inversa*

4) *Visualização dos sinais*: Os sinais amostrados são geralmente providos em um arquivo de dados. Muitas vezes, só a leitura de seus valores em uma tabela, por exemplo, não é suficiente para se extrair algum tipo de informação. Outros recursos devem estar disponíveis para melhor visualização. Gráficos são geralmente empregados por representarem de forma concisa e compacta um grande conjunto de dados.

Para a visualização de sinais provenientes de canais analógicos e digitais, gráficos representativos deverão ser gerados exibindo os valores em função do tempo. Existem outros tipos de canais, como sinais espectrais que são definidos no domínio da frequência. Também é importante a disponibilização de visualizações alternativas como a representação fasorial e o cálculo do valor RMS.

Outros tipos de recursos como a exibição de diversas curvas em um mesmo gráfico ou a exibição simultânea de gráficos distintos possibilitará a comparação entre curvas de forma mais amigável. A adição de cursores integrados em todos os gráficos também facilitará a comparação e visualização.

Outras formas de representação das curvas deverão estar disponíveis conforme as opções listadas a seguir:

- **Valor RMS**

Cada canal analógico ou derivado aritmético terá um gráfico com o valor RMS (*Root Mean Square*) correspondente. O valor RMS representa o valor quadrático eficaz da magnitude de uma quantidade variável.

- **Fasores**

Cada canal analógico ou derivado aritmético deverá ter uma série representando o ângulo e a amplitude máxima em um certo tempo. Essas séries devem ser mostradas em uma interface da própria janela interna de exibição de gráficos. Os controles deslizantes determinarão os tempos de representação do ângulo e da fase das séries.

5) *Multi-arquivos*: Muitas vezes é interessante trabalhar com canais que não estão em um mesmo arquivo, como dados provenientes de mais de uma estação. Por isso a capacidade do *software* em importar canais de mais de um arquivo por vez provê a flexibilidade necessária para comparação e manipulação cruzada.

O conceito de “multi-arquivos” deve ser aplicado a ferramenta, permitindo a manipulação de vários canais entre arquivos distintos de forma simultânea. A abertura de um arquivo não pode inviabilizar a visualização de canais previamente selecionados. Canais importados podem ser habilitados e desabilitados do sistema, dando maior flexibilidade e clareza nas manipulações.

6) *Tipos de canais*: O sistema deverá reconhecer três tipos de canais com suas características e formas de representação apresentadas a seguir. A figura 1 mostra a relação entre os diferentes tipos de canais.

- **Canais Analógicos**:

Canais cujos dados representam um sinal contínuo no tempo que foi discretizado para representação computacional.

O gráfico correspondente a esse tipo de canal deverá apresentar no eixo das abscissas o tempo em milissegundos e no eixo das ordenadas a unidade da série variante no

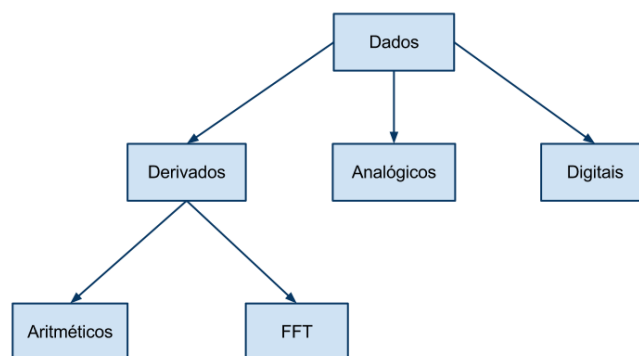


Figura 1: Tipos de Canais

tempo. Cada unidade possuirá um eixo no gráfico, e esses eixos são posicionados alternadamente entre a direita e a esquerda de cada gráfico.

- **Canais Digitais**:

Canais cujos dados representam um sinal discreto no tempo e binário, contendo apenas os bits “1” e “0”. Os canais digitais deverão ser representados por gráficos de categoria na horizontal. O eixo das abscissas deve representar o tempo em milissegundos e no eixo das ordenadas o nome do canal.

As barras desse tipo de gráfico devem possuir apenas duas cores: azul e vermelho. Cada cor representará um dos níveis lógicos do sinal digital. O azul corresponderá ao nível lógico “0” e o vermelho ao nível lógico “1”.

- **Canais Derivados**:

São canais gerados pelo próprio sistema através de operações do usuário. Os canais derivados ainda são divididos em duas subcategorias.

- **ARITMETICOS**: Canais gerados a partir de operações algébricas sobre canais analógicos (Canais aritméticos, em essência, são do mesmo tipo dos canais analógicos).

- **FFT**: Canais gerados a partir de uma transformada discreta de Fourier utilizando o algoritmo FFT sobre canais analógicos.

Os canais derivados FFT deverão ser representados por gráficos de categoria na vertical. O eixo das abscissas representará as harmônicas do sinal, enquanto no eixo das ordenadas a sua magnitude. Caso os dois canais tenham frequências fundamentais diferentes, um novo gráfico deverá ser criado para diferenciá-los, já que não podem dividir a multiplicidade das frequências harmônicas.

B. Requisitos não Funcionais

A adoção do programa só é totalmente realizada quando são atendidas a maioria das necessidades dos usuários e as diretivas de implementação, tais como: funcionalidade, velocidade, eficiência, confiabilidade, segurança, consistência, dentre outros.

1) *Usabilidade*: O usuário do sistema deve ter facilidade de uso, ou seja, conseguir ter uma visão geral da aplicação com poucas horas de treinamento. Também vários mecanismos devem estar disponíveis para facilitar o entendimento e

permitir ao usuário extrair de forma rápida e fácil as informações requeridas, como representações alternativas dos sinais amostrados (ex: valor eficaz, diagrama fasorial, transformadas de domínio, manipulações matemáticas, personalização da exibição de gráficos, etc).

Outro ponto importante detectado se refere à usabilidade e a interface do aplicativo, que deve ser simples e amigável sem engessar o comportamento do usuário, permitindo o foco na abordagem do problema e não na adaptabilidade da ferramenta.

2) *Portabilidade e código livre*: Com a crescente construção de sistemas computacionais heterogêneos é comum a adoção de diversos *softwares* e *hardwares* não compatíveis entre usuários. Grande exemplo disso é a variedade de arquiteturas adotadas em diferentes sistemas operacionais. Desta forma, o aplicativo deverá estar atento ao seu público alvo e ser capaz de atender a maioria de seu mercado potencial. A adoção de *software* livre permitirá que a própria comunidade de usuários se adapte às suas necessidades, trazendo benefícios para ambas as partes: por um lado, a comunidade de usuários da aplicação aumenta; por outro lado, a plataforma será beneficiada com a disponibilidade de mais aplicações.

3) *Ajuda*: Um sistema de ajuda integrado a ferramenta deve estar disponível ao usuário em todas as telas do sistema e na barra de tarefas na parte inferior várias dicas devem ser estar visíveis ao usuário de acordo com a ação e contexto do sistema.

4) *Velocidade*: Um sistema lento incomoda o usuário final e por isso otimizações devem ser pensadas desde a fase de projeto para que os esforços em processos críticos com grandes gastos de uso de processamento, memória ou acesso a disco sejam minimizados.

V. MODELAGEM DO *Software* PROPOSTO

Na fase de análise, a investigação das funcionalidades do aplicativo a ser desenvolvido, por meio do Documento de Requisitos gerado, possibilitou a construção do projeto de *Software* a ser desenvolvido. Os principais casos de uso estão representados através do diagrama de casos de uso da figura 2 que exemplifica os processos de abertura de arquivos, manipulação de canais analógicos e digitais a partir de operações matemáticas algébricas ou de transformadas resultando na criação de novos canais. Também é possível visualizar as formas de onda dos sinais, bem como escolher outros tipos de representação para cada canal. O resultado final pode ser salvo em um arquivo padrão de saída.

A tabela II ilustra como foi o modelo adotado para documentação dos casos de uso. Produziu-se um documento escrito para cada caso de uso, no qual especificou-se de forma detalhada a sequência típica do fluxo de informações, pré e pós condições, além de sequências alternativas. A tabela II exemplifica a descrição do caso de uso *Exibir Gráficos*.

De posse das informações contidas nos casos de uso foi possível construir um modelo de domínio parcial para a aplicação. Um modelo de domínio permite visualizar e relacionar conceitos de domínio, além de ilustrar uma abstração das classes conceituais.

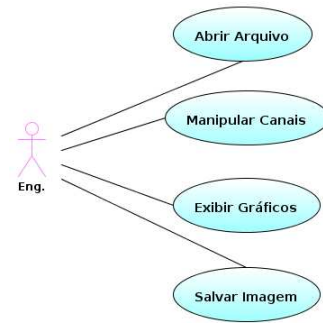


Figura 2: Diagrama de casos de uso

Tabela II: Caso de uso: Exibir Gráficos

Caso de Uso	Exibir Gráficos
Ator Principal	Engenheiro de proteção ou de qualidade
Interessados e Interesses	Engenheiro de proteção ou de qualidade, gerentes e diretores de companhias elétricas
Pré-condições	<ul style="list-style-type: none"> • Abertura do respectivo arquivo contendo os dados oscilográficos a serem analisados • Seleção dos canais a serem exibidos ou a criação de canais a partir de manipulações matemáticas
Pós-Condições	Curvas geradas e dispostas em áreas compartilhadas ou individuais com as respectivas representações selecionadas
Fluxo Básico	<ol style="list-style-type: none"> 1) Usuário configura áreas de exibição de acordo com as restrições de conjugação 2) <i>Software</i> gera gráficos de acordo com configuração selecionada 3) Usuário muda formas de representação de forma a adequar o tipo de informação a ser extraída 4) Fim do caso de uso

O domínio parcial do aplicativo é apresentado na figura 5. Por essa, nota-se que o sistema WAPS central é responsável por representar o conjunto de dados amostrados através de uma estrutura padrão independente do formato de entrada do arquivo lido. Esse formato padrão é dividido em duas partes. A primeira é responsável por armazenar informações de configurações como nome, tipo e quantidade de canais. A segunda cuida das amostras dos sinais de entrada.

As manipulações algébricas são realizadas pelo módulo *Math Builder* que além de operações algébricas e auxiliares comuns também disponibiliza operações de transformadas no domínio da frequência, bem como a sua inversa.

O WAPS trabalha com o conceito de multi-arquivos provendo a flexibilidade necessária para comparação e manipulação cruzada além de disponibilizar a seleção de canais mudando, assim, o escopo de visibilidade de cada arquivo.

Durante a fase de projeto foram desenvolvidos os diagramas de sequência para os principais casos de uso. Na figura 4 as trocas de mensagens entre os objetos são mostradas durante uma operação matemática realizada via *Math Builder*.

Além de uma visão dinâmica de objetos, mostrada na figura 4, é útil criar uma visão estática das definições de classes com um diagrama de classes de projeto [11]. A figura 5 exemplifica

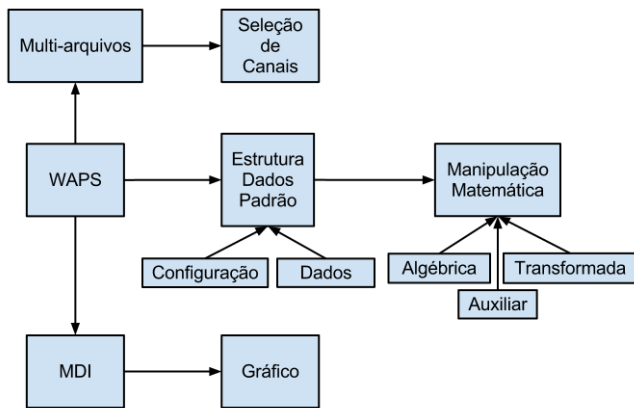


Figura 3: Modelo de domínio parcial

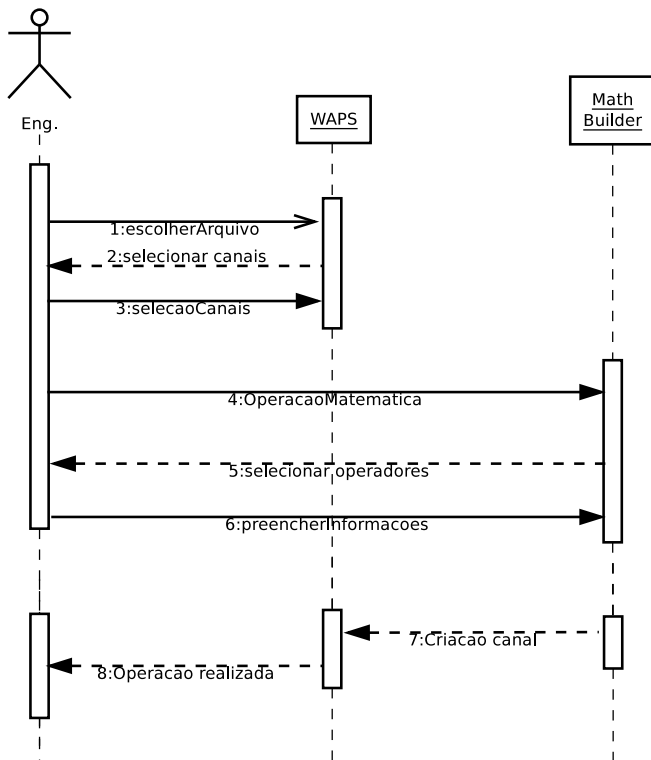


Figura 4: Diagrama de sequência para operações matemáticas

algumas das classes projetadas.

Nota-se que existe uma interface denominada *Padrão* que não possui atributos e fornece um contrato de implementação de métodos a fim de resgatar informações dos sinais elétricos importados no sistema. As classes *COMTRADE91* e *COMTRADE99* implementam tal interface sendo que a última é uma atualização da primeira e por isso herda suas características.

Os tipos de canais foram divididos entre as classes *Analogico*, *Digital* e *Derivado* que herdam as propriedades da classe *Canal*, cada uma possuindo atributos e métodos específicos.

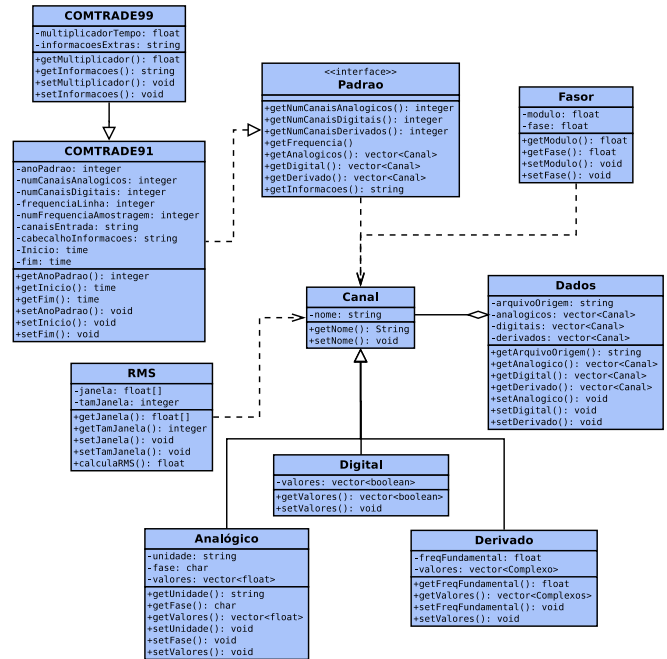


Figura 5: Diagrama de classes parcial

VI. IMPLEMENTAÇÃO DO *Software* PROPOSTO

A fim de se construir um artefato computacional modular e flexível, optou-se pela adoção do paradigma de Programação Orientado a Objetos (POO) [12], devido aos vários apontamentos encontrados na literatura [6], [13]–[16] descrevendo sua utilização como uma excelente alternativa para o desenvolvimento de aplicações voltadas para o setor elétrico [17].

O WAPS se encontra atualmente em versão beta, através do link: WAPS. Em particular, para a sua implementação, foi escolhida a linguagem de programação Java baseada nos preceitos de POO, por ser uma linguagem de alto nível, flexível e amplamente utilizada comercialmente. Além disso, a mesma fornece a portabilidade necessária para execução do aplicativo em vários sistemas operacionais e dá suporte ao incremento de módulos para reconhecimento de outros padrões, a partir dos conceitos de Hierarquia e Encapsulamento [18].

Dentre os padrões de registros de oscilografias existentes, escolheu-se o padrão IEEE COMTRADE como o formato inicial para a manipulação dos dados devido ao seu grande uso tanto no meio acadêmico, quanto no meio industrial. O *software* interpreta os dados a partir desta estrutura padrão e para o reconhecimento de outros padrões é necessário apenas o desenvolvimento de módulos capazes de transformar o formato de entrada para o formato base.

O *software* foi desenvolvido em dois módulos conforme representado na figura 6. O módulo 1 é responsável por construir a infra-estrutura base do *software*, enquanto o Módulo 2 é responsável pela visualização e análise dos dados.

O módulo 1 é capaz de importar um arquivo oscilográfico escolhido pelo usuário e convertê-lo para o formato base. Esse formato comum contém as informações mínimas necessárias para a realização das operações algébricas em cada instante de tempo da amostra, além de valores para o cálculo da

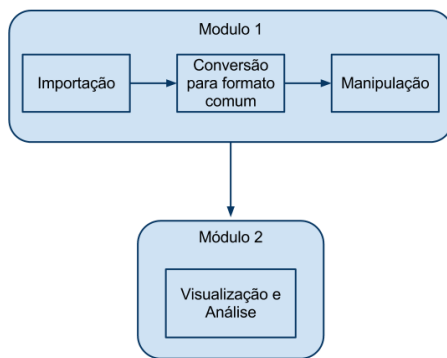


Figura 6: Módulos do *software* WAPS

Transformada Rápida de Fourier, Transformada inversa de Fourier, dos valores RMS e fasoriais do sinal.

Já o módulo 2 é responsável pela exibição dos sinais importados e criados através das manipulações realizadas pelo módulo 1. Desta forma, é possível visualizar de maneira clara os valores das amostras, além das representações alternativas como a aplicação da FFT, dos valores eficazes e da representação fasorial. Também é possível realizar comparações entre os sinais gerados.

VII. TESTES REALIZADOS

Testes foram realizados a partir de um projeto bem definido e que engloba a maioria das funcionalidades da ferramenta, tais como: a utilização do conceito de multi arquivos, uso de canais analógicos e digitais, geração de arquivos derivados aritméticos e de transformada, exibição de valores instantâneos e RMS, etc.

Para exemplificar e analisar a confiabilidade da ferramenta, o projeto de testes foi dividido em duas fases conforme explanados nas próximas seções.

A. Projeto de testes: Fase 1

O projeto da fase 1 pode ser visualizado na figura 7. Dois arquivos são selecionados. O primeiro arquivo possui tanto arquivos analógicos quanto digitais, mas foram selecionados apenas 4 canais analógicos (canais de tensão: *UL1E*, *UL2E* e *UL3E*; canais de corrente: *IL2* e *IL3*) e 2 canais digitais (*Start* e *Trip*) de seu repositório. O segundo arquivo possui apenas canais analógicos e serão selecionados 3 destes canais (*BVA1MA*, *BVA1MC* e *SUBA*). Em seguida, foram realizadas duas operações aritméticas: soma entre dois canais analógicos (canais *UL1E* e *UL2E*) e “negativação” da amostra (canal *SUBA*). Para verificação dos resultados obtidos são mostrados os gráficos representativos de cada canal.

A figura 8 mostra os valores instantâneos dos canais *UL1E* (vermelho), *UL2E* (azul) e do canal criado *UL1E+UL2E* (verde). Pode-se observar que o canal *UL1E+UL2E* realmente representa a soma dos outros dois canais que estão defasados. Ele segue os canais ficando entre ambos. Perto do tempo de 110 ms, após o começo da amostragem, é possível ver a distorção do sinal *UL1E* ocorrida devida a uma falta elétrica. Como seu valor diminui

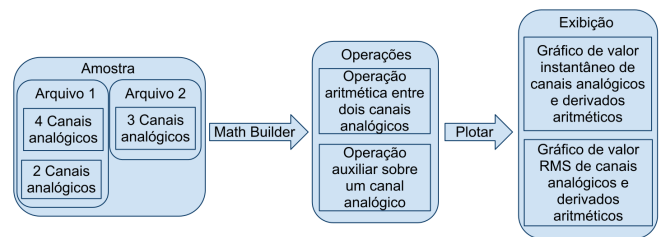


Figura 7: Planejamento de teste: Fase 1

em relação ao canal *UL2E*, o canal soma possui valores próximos a esse último canal.

Os canais *Start* e *Trip* são canais digitais e por isso possuem representações em formato de barras horizontais. Canais digitais geralmente representam eventos lógicos que ocorrem na linha de transmissão. Já a figura 9 mostra a relação entre o sinal *SUBA* e seu respectivo valor negativo *-SUBA*.

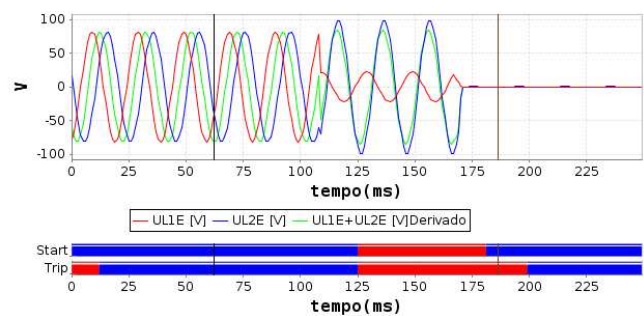


Figura 8: Fase 1: Canais analógicos, aritméticos e digitais

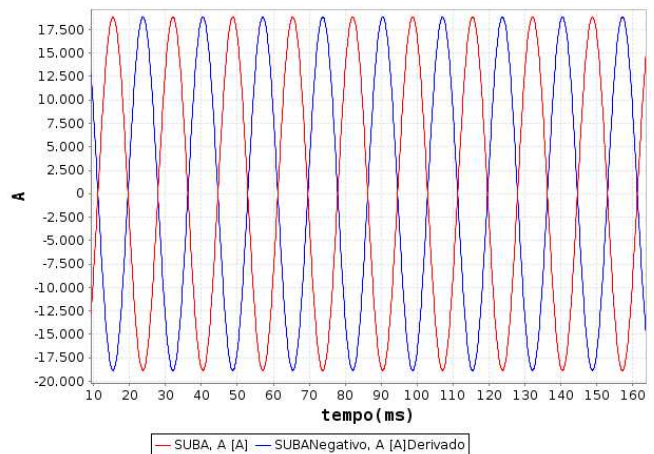


Figura 9: Fase 1: "Negativação" do canal analógico

A figura 10 é a representação dos canais em valores RMS. Pode-se perceber que o valor eficaz de todos os canais antes de ocorrer a falta se mantém constante e com mesmo valor. Isso ocorre porque os canais *UL1E* e *UL2E* estão defasados e como a soma entre ambos possui mesma amplitude o valor eficaz se mantém constante. Após a falta, o sinal *UL1E* cai em relação ao valor do canal *UL2E* e, portanto, o canal soma possui valor muito próximo do canal *UL2E*.

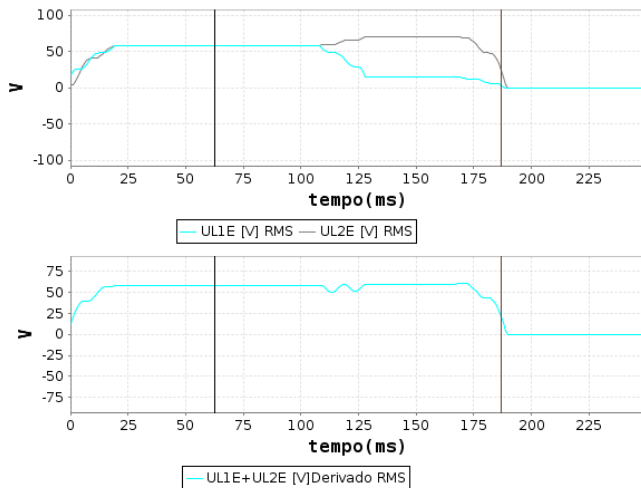


Figura 10: Fase 1: Valores RMS dos canais analógicos

B. Projeto de testes: Fase 2

O projeto da fase 2 pode ser visualizado na figura 11. Nessa fase apenas um arquivo, contendo canais analógicos, é selecionado. Aqui apenas um canal será selecionado (*BVA1MA*) para a aplicação da transformação rápida de Fourier. Para verificação dos resultados obtidos, os gráficos representativos do sinal original e transformado são apresentados.

A figura 12 aglutina os gráficos dos dois canais analisados. O primeiro gráfico diz respeito ao próprio sinal analógico *BVA1MA*. É uma representação no domínio do tempo. O segundo gráfico diz respeito a transformada FFT sobre esse canal. É uma representação no domínio da frequência e, portanto, apresenta o valor DC do sinal, bem como suas harmônicas nas frequências múltiplas da frequência fundamental.

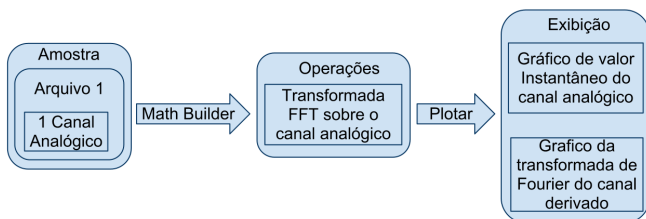


Figura 11: Planejamento de teste: Fase 2

Pode-se perceber que a visualização dos dados é diferenciada para cada tipo de canal e, para a maioria dos canais, existe mais de um tipo de representação, ajudando o usuário da aplicação a obter o maior número de informações de forma rápida e fácil. Para aumentar a flexibilidade, a tela de configuração permite ao usuário definir como serão exibidas as diversas informações, podendo-se agrupá-las da forma requerida.

Canais são diferenciados por cor permitindo a distinção imediata de suas formas. As diversas legendas são separadas por gráfico e representam seus respectivos canais, indicando suas unidades de medidas que são expostas nos eixos coordenados.

Os resultados obtidos revelam a concordância com o valor esperado para as operações propostas, indicando a viabilidade

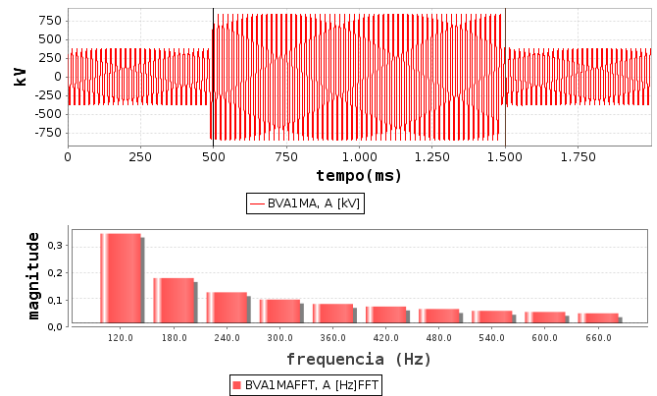


Figura 12: Fase 2: Valor instantâneo e representação espectral

da ferramenta. O teste revela a capacidade da ferramenta em um ambiente multi arquivos com canais distintos.

VIII. CONCLUSÕES

Este trabalho aborda a construção de uma nova ferramenta, denominada WAPS (*Wave Analyser for Power System*), voltada ao estudo e análise de perturbações e transitórios de redes de energia elétrica. Foram utilizados preceitos de *software* livre, permitindo à comunidade científica, voltada à área de proteção de sistemas elétricos de potência, utilizá-la de forma gratuita e adaptá-la às suas necessidades.

O uso dos conceitos de orientação a objetos permitiu construir um *software* modular e bem organizado, ajudando possíveis desenvolvedores que queiram modificá-lo. Além disso, a independência do formato de entrada permite o reconhecimento e adaptabilidade de diversos tipos de arquivos sem que haja modificações no núcleo do código. A utilização da linguagem Java permitiu a portabilidade essencial para disseminar a ferramenta em diversos ambientes com arquiteturas distintas.

A ferramenta possui uma interface simples e amigável, com sistema de ajuda integrado. Cada tipo de canal possui representação própria, permitindo melhor visualização e extração de suas características intrínsecas.

Desta forma, procurou-se desenvolver uma ferramenta que atendesse as necessidades do mercado, agregando ideias novas às funcionalidades já existentes, além de construir um produto final gratuito e flexível. Os autores esperam ter atingido os objetivos propostos e aberto novas oportunidades para extensão e continuidade do projeto.

AGRADECIMENTOS

Os autores agradecem o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo financiamento prestado, bem como ao Laboratório de Sistemas de Energia Elétrica (LSEE) da Escola de Engenharia Elétrica de São Carlos (EESC/USP) pela infra-estrutura fornecida durante o desenvolvimento deste.

REFERÊNCIAS

- [1] C. PES, "IEEE standard common format for transient data exchange (comtrade) for power systems," *IEEE Std C37.111-1991*, p. 1, 1991.
- [2] PES, "IEEE standard common format for transient data exchange (comtrade) for power systems," *IEEE Std C37.111-1999*, p. i, 1999.
- [3] P. Barcellos, C. Oliveira, and M. Anido, "A novel waveform analyzer for analog and digital signals for the windows(r) environment," in *Circuits and Systems, 1996., IEEE 39th Midwest symposium on*, vol. 1, aug 1996, pp. 385 –388 vol.1.
- [4] G. Kiessling and S. Schwabe, "Software solution for fault record analysis in power transmission and distribution," in *Developments in Power System Protection, 2004. Eighth IEE International Conference on*, vol. 1, april 2004, pp. 192 – 195 Vol.1.
- [5] Q. Dam and A. Meliopoulos, "Relay simulation and testing software on the .net framework environment," in *Power Engineering Society General Meeting, 2007. IEEE*, june 2007, pp. 1 –5.
- [6] M. N. Agostini, I. C. Decker, and A. S. Silva, "Desenvolvimento e implementação de uma base computacional orientada a objetos para aplicações em sistemas de energia elétrica," *Sba: Controle & Automação Sociedade Brasileira de Automática*, vol. 13, pp. 181–189, 2002.
- [7] D. Bobrow, *Perspective on Artificial Intelligence Programming*, ser. Artificial intelligence. MIT Press, 1994.
- [8] G. Booch, *Object Oriented Analysis & Design With Application*. Pearson Education, 2006.
- [9] P. Deitel and H. Deitel, *Java How to Program: Late Objects Version*, ser. How to program series. Pearson/Prentice Hall, 2009.
- [10] M. Blaha and J. Rumbaugh, *Object - Oriented Modeling And Design With Uml, 2/E*. Pearson Education, 2007.
- [11] R. A. L. Rabêlo, W. N. M. Lopes, A. A. F. M. Carneiro, and R. T. V. Braga, "Análise, projeto e implementação orientados a objetos aplicados ao planejamento da operação energética de sistemas hidrotérmicos de potência," *Congresso Brasileiro de Automática - CBA*, 2006.
- [12] C. Larman, *Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos*. Bookman, 2002.
- [13] X. Qiu and W. Wimmer, "Applying object-orientation and component technology to architecture design of power system monitoring," in *Power System Technology, 2000. Proceedings. PowerCon 2000. International Conference on*, vol. 2, 2000, pp. 589 –594 vol.2.
- [14] A. Losi and M. Russo, "Dispersed generation modeling for object-oriented distribution load flow," *Power Delivery, IEEE Transactions on*, vol. 20, no. 2, pp. 1532 – 1540, april 2005.
- [15] T. McDermott, "An open source distribution system simulator," in *Power Engineering Society General Meeting, 2006. IEEE*, 0-0 2006, p. 4 pp.
- [16] D. Peharda, Z. Hebel, and A. Barta, "Power system topology assessment and pre-estimation in an object oriented environment," in *AFRICON 2007*, sept. 2007, pp. 1 –7.
- [17] H. M. C. Branco, R. M. Monaro, D. Barbosa, U. C. Netto, M. Oleskovicz, D. V. Coury, S. E. Tokuno, and O. R. M. Parreira, "Processo unificado aplicado ao projeto e desenvolvimento de um software compatível as análises de proteção e qualidade da energia elétrica em um sistema de distribuição real," *CBA: Congresso Brasileiro de Automática*, vol. 18, 2010.
- [18] D. Poo, D. Kiong, and S. Ashok, *Object-Oriented Programming and Java*. Springer, 2007.

BIOGRAFIAS

Athila Quaresma Santos possui graduação em Engenharia de Computação pela Universidade de São Paulo – USP (São Carlos - 2011) sendo bolsista de iniciação científica por 2 anos. Atualmente é aluno regular e bolsista CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) do programa de pós-graduação em engenharia elétrica da Universidade de São Paulo – Campus São Carlos, a nível de doutorado, se aperfeiçoando na área de Sistemas Elétricos de Potência, *Smart Grids* e sistemas inteligentes.

Diego de Castro Grillo é aluno de graduação em Engenharia de Computação pela Universidade de São Paulo – USP (São Carlos). Realizou iniciação científica com bolsa FUNDEB (2009-2010). Tem experiência na área de Engenharia Elétrica, com ênfase em computação. Atualmente realiza estágio na empresa Itaú de Jaguariúna.

Ulisses Chemin Netto graduado em Engenharia Industrial Elétrica pela Universidade Tecnológica Federal do Paraná (2003) e titulado como Mestre em Engenharia Elétrica, na sub-área de Sistemas Elétricos de Potência, pela Escola de Engenharia de São Carlos, da Universidade de São Paulo (2008). Atualmente é doutorando na mesma instituição na qual obteve seu título de mestre em Engenharia. Suas principais áreas de interesse incluem a proteção de sistemas elétricos de potência e processos de ensino/aprendizagem em Engenharia.

Daniel Barbosa possui graduação em Engenharia Elétrica pela Universidade Federal da Bahia (2005), mestrado em Engenharia Elétrica pela Universidade de São Paulo (2007) e doutorado em Engenharia Elétrica pela Universidade de São Paulo (2010). Atualmente é professor da Universidade Salvador e professor adjunto da Universidade Federal da Bahia. Tem experiência na área de Engenharia Elétrica, com ênfase em Medição, Controle, Correção e Proteção de Sistemas Elétricos de Potência, atuando principalmente nos seguintes temas: ATP, proteção diferencial, sistemas elétricos de potência, qualidade da energia elétrica e lógica fuzzy.

Denis Vinicius Coury possui graduação em engenharia elétrica pela Universidade Federal de Uberlândia (1983), mestrado em engenharia elétrica pela Universidade de São Paulo (1987) e doutorado em Electrical Engineering na Bath University- UK (1992). Passou o seu ano Sabático na Cornell University-USA em 1999/2000. Atualmente é professor Titular da Universidade de São Paulo. Tem experiência na área de Engenharia Elétrica, com ênfase em Proteção Digital de Sistemas Elétricos de Potência, atuando principalmente nos seguintes temas: Proteção Digital Adaptativa, Aplicação de Ferramentas Inteligentes em Sistemas Elétricos de Potência, Localização de Falhas em Linhas de Distribuição e Transmissão, Simulação Computacional de Sistemas Faltosos e, mais recentemente, Qualidade da Energia Elétrica.

Mário Oleskovicz possui graduação em engenharia elétrica pela Universidade Federal de Santa Catarina – UFSC (1995), obtendo os títulos de mestre e de doutor em engenharia elétrica pela Escola de Engenharia de São Carlos – EESC, Universidade de São Paulo – USP, em 1997 e 2001, respectivamente. Atualmente é professor doutor da Universidade de São Paulo. Tem experiência na área de engenharia elétrica – sistemas elétricos de potência, com ênfase nas subáreas de proteção digital de sistemas elétricos de potência e qualidade da energia elétrica.

Ferramenta Multi Arquivos para Visualização e Manipulação Oscilográfica de Sinais Elétricos

FERRAMENTA MULTI ARQUIVOS PARA VISUALIZAÇÃO E MANIPULAÇÃO OSCILOGRÁFICA DE SINAIS ELÉTRICOS

ATHILA Q. SANTOS*, DIEGO C. GRILLO*, ULISSES C. NETTO*, DANIEL BARBOSA†, DENIS V. CORY*, MÁRIO OLESKOVICZ*

**Departamento de Engenharia Elétrica, Escola de Engenharia de São Carlos
Universidade de São Paulo (USP)
Av. Trabalhador Sancarlense, 400, São Carlos, São Paulo, Brasil*

†*Mestrado em Energia
Universidade Salvador (UNIFACS)
Rua Ponciano de Oliveira, 126, 2 andar, Rio Vermelho, Salvador, Bahia, Brazil*

Emails: athila.santos@usp.br, diego.grillo@usp.br, uchenin@usp.br,
daniel.barbosa@pro.unifacs.br, cory@sc.usp.br, olesk@sc.usp.br

Abstract— With the massive increase in the amount of data related to electric power systems, the need for well-defined formats for storing and exchanging information is evident. Another important factor is the construction of well-structured and flexible software for the analysis of events and information extraction in such systems, certifying that it will be made quickly and correctly. Thus, an open source tool called WAPS (*Wave Analyser for Power System*) was developed in this paper for study and analysis of transient disturbances in power grids. As a result, a product with very flexible interface was obtained. It can also have many different kinds of inputs. The tool also allows working with several files simultaneously with oscillographic representation for each distinct type of channel. Many algebraic operations on the signals are available, enabling the creation of new channels and allowing options such as the representation of phasors or RMS values, providing greater readability for the information provided.

Keywords— WAPS, COMTRADE, Oscillography, Protection, Electrical Power Systems.

Resumo— Com o aumento massivo da quantidade de dados relacionados aos Sistemas Elétricos de Potência surge a necessidade de formatos bem definidos para o armazenamento e intercâmbio das informações. Outro fator importante é a construção de *softwares* bem estruturados e flexíveis para que a análise de eventos e extração de informações em tais sistemas sejam feitas de forma rápida e correta. Desta forma, uma ferramenta em código livre denominada WAPS (*Wave Analyser for Power System*) foi desenvolvida voltada ao estudo e análise de perturbações e transitórios de redes de energia elétrica. Como resultado, obteve-se um produto com interface intuitiva e bastante flexível capaz de comportar diversos formatos de entrada. A ferramenta ainda permite o trabalho simultâneo entre diversos arquivos oscillográficos com representação distinta para cada tipo de canal. Varias formas de manipulações algébricas sobre sinais estão disponíveis possibilitando a criação de novos canais além de permitir opções de visualização usuais como a representação fasorial ou RMS provendo maior legibilidade das informações.

Palavras-chave— WAPS, COMTRADE, Oscilografia, Proteção, Sistemas Elétricos de Potência.

1 Introdução

O aumento do uso de tecnologia digital em dispositivos de proteção, medição e controle, principalmente em subestações de energia elétrica, resultou no acúmulo de grandes quantidades de registros digitais nas mais diversas estruturas, muitas das quais, em formatos proprietários sem qualquer tipo de padronização (Committee, 1991).

Desta forma, a integração entre os diferentes processos de análise e de estudos, nos vários setores da indústria elétrica, é complexa e necessita do conhecimento de múltiplas ferramentas computacionais. Assim, com a necessidade do intercâmbio de informações para melhorar e automatizar procedimentos, surge a conveniência em se adotar um formato unificado e padronizado para registros digitais (Committee, 1999).

Nesse sentido, o *Institute of Electrical and Electronics Engineers* (IEEE) define um formato comum para arquivos de dados digitais e meio

de troca para intercâmbio de vários tipos de faltas, testes e simulações, sendo capaz de representar formas de ondas geradas a partir de registradores de faltas, relés de proteção, medidores de qualidade de energia e dispositivos similares. Tal padrão é conhecido como COMTRADE, cujo acrônimo significa *COMmon format for TRANsient Data Exchange*, e é amplamente adotado tanto pela indústria quanto pela academia.

Outros padrões existentes no mercado como o PQDIF, *Power Quality Data Interchange Format*, também recomendado pelo IEEE para a troca de dados referentes à qualidade de energia (Committee, 2004), é uma alternativa viável para registros relacionados à área de sistemas elétricos. Embora exista uma sobreposição de nichos entre o padrão COMTRADE e PQDIF, ambos foram otimizados para diferentes propósitos (King and Gunther, 2006).

Com a padronização dos arquivos, diversos aplicativos foram desenvolvidos com o objetivo de

visualizar as formas de onda contidas em registros oscilográficos, bem como facilitar a análise destas. Todavia, a maioria destas ferramentas são comerciais e não estão acessíveis a todos os profissionais do setor elétrico. Tal fato, motivou o desenvolvimento de uma ferramenta gratuita, livre (*open source*) e que pudesse ser utilizada em quaisquer sistemas operacionais.

Outro ponto importante detectado se refere à usabilidade e a interface desses aplicativos que muitas vezes engessam o comportamento do usuário, que deixa de fixar no problema abordado para se adequar aos mecanismos do sistema. A adoção do programa só é totalmente realizada quando são atendidas a maioria das necessidades dos usuários e as diretivas de implementação, tais como: funcionalidade, velocidade, eficiência, confiabilidade, segurança, consistência, dentre outros.

Neste contexto, este trabalho apresenta a ferramenta computacional WAPS (*Wave Analyser for Power System*), desenvolvida com todas as suas características e funcionalidades disponíveis em diversas plataformas, sendo capaz de tornar os dados referentes aos fenômenos ligados à geração, transmissão e distribuição de energia elétrica manipuláveis computacionalmente. Testes de desempenho com arquivos oscilográficos provenientes de equipamentos comerciais e de simulações computacionais foram realizados e como resultado, obteve-se um aplicativo com interface amigável que permite ao usuário rápido aprendizado além de fornecer um sistema de ajuda integrado. Sequências simples e consistentes de interação mostram claramente as alternativas disponíveis a cada passo.

2 Funcionalidades

É possível encontrar na literatura diversos aplicativos, em sua maioria comerciais, capazes de realizar manipulações ou exibir informações de arquivos oscilográficos (Barcellos et al., 1996; Kiessling and Schwabe, 2004; Dam and Meliopoulos, 2007). Dentre os mais utilizados, pode-se citar o SINAPE (Sistema Integrado de apoio à Análise de Perturbações) desenvolvido pelo Centro de Pesquisas de Energia Elétrica da Eletrobrás (CEPEL). Outro aplicativo é o TOP (*The Output Processor*), desenvolvido pela Electrotek Concepts, permite a visualizações dos dados de diversos formatos, entre eles o COMTRADE. O *software* SIEMENS SIGRA é uma ferramenta que reconhece apenas arquivos no formato COMTRADE e está disponível apenas para o sistema operacional Windows, sendo que sua licença não é gratuita. Já o .NET Relayer é um programa simples, anteriormente denominado *Waveform Analyser*, é um visualizador de arquivos COMTRADE cujos cálculos matemáticos precisam ser programados antes de serem exibidos, o que pode ser uma vantagem para cál-

culos específicos, porém um problema no caso de manipulações muito usuais.

Espera-se que um bom *software* especializado nesse tipo de aplicação forneça mecanismos para que seus usuários possam extrair, de forma rápida e fácil, as informações requeridas. Dentre as funcionalidades comumente conhecidas por profissionais da área de SEP pode-se citar:

1. Reconhecimento de diversos formatos de entrada:

Softwares especializados devem reconhecer os principais formatos de arquivo usados em sua área de atuação. Também devem ser capazes de se adaptarem a possíveis atualizações de formatos, bem como o surgimento de novos padrões. Uma arquitetura modular facilita a incorporação de diferentes funcionalidades mesmo após o lançamento do aplicativo.

2. MDI:

Interface de documentos múltiplos ou MDI (do inglês *Multiple Document Interfaces*) é um método de organização de aplicações gráficas em painéis que residem em uma janela principal.

3. Portabilidade e código livre:

Com a crescente construção de sistemas computacionais heterogêneos é comum a adoção de diversos *softwares* e *hardwares* não compatíveis entre usuários. Um grande exemplo disso é a variedade de arquiteturas adotadas em diferentes sistemas operacionais. Desta forma, um aplicativo deve estar atento ao seu público alvo e ser capaz de atender a maioria de seu mercado potencial. Para isso, a construção de aplicativos multiplataformas são práticas cada vez mais comuns, a partir de modelagens modelares e linguagens com suporte às várias arquiteturas existentes.

A adoção de *software* livre permite que a própria comunidade de usuários se adapte às suas necessidades, trazendo benefícios para ambas as partes: por um lado, a comunidade de usuários da aplicação aumenta; por outro lado, a plataforma é beneficiada com a disponibilidade de mais aplicações.

4. Operações Matemáticas:

Operações matemáticas auxiliam na manipulação dos sinais permitindo aos usuários realizarem as modificações necessárias para a análise de interesse. Dentre as operações matemáticas mais comuns estão: operações algébricas, cálculo da transformada de fourier ou FFT (do inglês *Fast Fourier Transform*), transformada inversa de fourier ou IFT (do inglês *Inverse Fourier Transform*), cálculo dos valores eficazes ou RMS (do inglês *Root Mean Square*), sincronização entre ondas, etc.

5. Visualização dos Sinais:

Os sinais amostrados são geralmente providos em um arquivo de dados. Muitas vezes, só a leitura de seus valores em uma tabela, por exemplo, não é suficiente para se extrair algum tipo de informação. Outros recursos devem estar disponíveis para melhor visualização. Gráficos são geralmente empregados por representarem de forma concisa e compacta um grande conjunto de dados.

Para a visualização de sinais provenientes de canais analógicos e digitais, gráficos representativos devem ser gerados exibindo os valores em função do tempo. Existem outros tipos de canais, como sinais espectrais que são definidos no domínio da frequência. Também é importante a disponibilização de visualizações alternativas como a representação fasorial e o cálculo do valor RMS.

Outros tipos de recursos como a exibição de diversas curvas em um mesmo gráfico ou a exibição simultânea de gráficos distintos possibilita a comparação entre curvas de forma mais amigável. A adição de cursores integrados em todos os gráficos também facilita a comparação e visualização.

6. Algoritmos:

É comum a integração de algoritmos clássicos ligados a área de SEP como a localização de faltas, cálculo de componente simétricas, impedâncias, análise harmônica, etc.

7. Exportação:

A opção de exportação permite a conversão de um formato de entrada em um formato de saída desejado. Ainda é possível a geração de um novo arquivo se o aplicativo possibilitar a geração ou modificação de canais.

8. Multi arquivos:

Muitas vezes é interessante trabalhar com canais que não estão em um mesmo arquivo, como dados provenientes de mais de uma estação. Por isso, a capacidade do *software* em importar canais de mais de um arquivo por vez provê a flexibilidade necessária para comparação e manipulação cruzada.

Embora a maioria das funcionalidades mencionadas existam nos *softwares* existentes no mercado, o conjunto proposto não é encontrado em um único aplicativo, viabilizando a construção de uma ferramenta de código aberto que se adapte aos diversos tipos de canais existentes, tanto na norma COMTRADE, quanto aos novos canais criados dentro de seu domínio e que possuam representações distintas. Também deve permitir a manipulação de diversos arquivos e formatos, trabalhar com cursores e gráficos simultâneos, além

de possibilitar o uso de diversas janelas em uma mesma área de trabalho (MDI).

3 Arquitetura do Software

A fim de se construir um artefato computacional modular e flexível, optou-se pela adoção do paradigma de Programação Orientado a Objetos (POO) (Larman, 2002), devidos aos vários apontamentos encontrados na literatura (Qiu and Wimmer, 2000; Agostini et al., 2002; Losi and Russo, 2005; McDermott, 2006; Peharda et al., 2007) descrevendo sua utilização como uma excelente alternativa para o desenvolvimento de aplicações voltadas para o setor elétrico (Branco et al., 2010).

Em particular, para a implementação da ferramenta WAPS foi escolhida a linguagem de programação Java baseada nos preceitos de POO, por ser uma linguagem de alto nível, flexível e amplamente utilizada comercialmente. Além disso, a mesma fornece a portabilidade necessária para execução do aplicativo em vários sistemas operacionais e dá suporte ao incremento de módulos para reconhecimento de outros padrões, a partir dos conceitos de Hierarquia e Encapsulamento (Poo et al., 2007).

Dentre os padrões de registros de oscilografias existentes, escolheu-se o padrão IEEE COMTRADE como o formato inicial para a manipulação dos dados devido ao seu grande uso tanto no meio acadêmico, quanto no meio industrial. Embora esse formato tenha sido adotado para a implementação das diversas funcionalidades, é importante ressaltar que os dados lidos são armazenados em uma estrutura de dados geral, a qual independe do tipo de arquivo de entrada. O *software* interpreta os dados a partir de uma estrutura padrão e para o reconhecimento de outros padrões é necessário apenas o desenvolvimento de módulos capazes de transformar o formato de entrada para o formato base.

O *software* foi desenvolvido em dois módulos conforme representado na figura 1. O módulo 1 é responsável por construir a infra-estrutura base do *software*, enquanto o Módulo 2 é responsável pela visualização e análise dos dados.

O módulo 1 é capaz de importar um arquivo oscilográfico escolhido pelo usuário e convertê-lo para o formato base. Esse formato comum contém as informações mínimas necessárias para a realização das operações algébricas em cada instante de tempo da amostra, além de valores para o cálculo da Transformada Rápida de Fourier, Transformada inversa de Fourier, dos valores RMS e fasoriais do sinal.

Já o módulo 2 é responsável pela exibição dos sinais importados e criados através das manipulações realizadas pelo módulo 1. Desta forma, é possível visualizar de maneira clara os valores

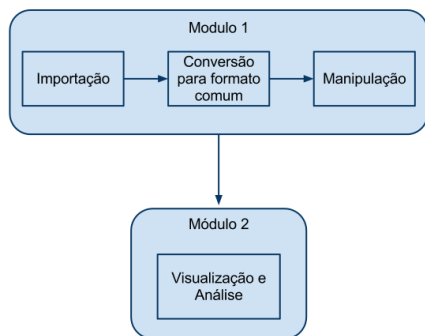


Figura 1: Módulos do *software* WAPS

das amostras, além das representações alternativas como a aplicação da FFT, dos valores eficazes e da representação fasorial. Também é possível realizar comparações entre os sinais gerados.

4 WAPS: Um *Software* Aplicado a Análise de Sinais Provenientes do Sistema Elétrico de Potência

O WAPS permite a visualização e manipulação de diversos tipos de sinais, bem como a criação de novos canais a partir dos dados importados. Além disso, é possível utilizar vários ambientes de trabalho na mesma aplicação, conforme os preceitos de MDI. Algumas das características do sistema são descritas a seguir:

- Manipulação de vários arquivos simultaneamente;
- Compatibilidade de arquivos COMTRADE binário e ASCII;
- Suporte a vários tipos de canais (Analogicos, Digitais, etc.);
- Gráficos diferenciados para cada tipo de canal;
- Visualização de sinais pelo valor médio RMS;
- Visualização espectral dos sinais;
- Visualização de formas de onda através de faixas;
- Várias áreas para visualização de gráficos (MDI);
- Múltiplas curvas por gráfico;
- Operações algébricas sobre sinais.

4.1 Tipos de canais

Existem três tipos de canais no sistema, conforme pode ser visualizado na figura 2. Cada tipo de canal possui sua respectiva representação. A seguir são apresentadas suas características.

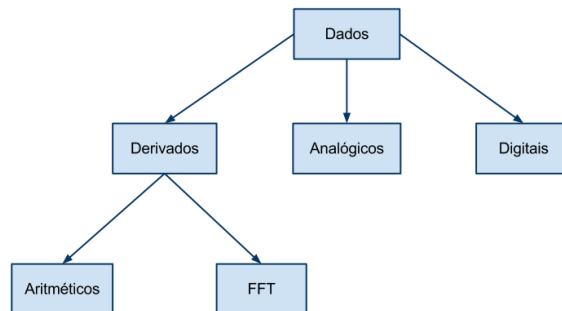


Figura 2: Tipos de Canais

1. Canais Analógicos:

Canais cujos dados representam um sinal contínuo no tempo que foi discretizado para representação computacional.

O gráfico correspondente a esses canais apresenta no eixo das abscissas o tempo em milissegundos e o eixo das ordenadas representa a unidade da série variante no tempo. Cada unidade possuirá um eixo no gráfico, e esses eixos são posicionados alternadamente entre a direita e a esquerda do gráfico.

2. Canais Digitais:

Canais cujos dados representam um sinal discreto no tempo e binário, contendo apenas os bits “1” e “0”.

Os canais digitais são representados por gráficos de categoria na horizontal. O eixo das abscissas representa o tempo em milissegundos. O eixo das ordenadas apresenta o nome do canal.

As barras desse tipo de gráfico possuem apenas duas cores: azul e vermelho. Cada cor representa um dos níveis lógicos do sinal digital. O azul corresponde ao nível lógico “0” e o vermelho ao nível lógico “1”.

3. Canais Derivados:

São canais gerados pelo próprio sistema através de operações do usuário. Os canais derivados ainda são divididos em duas subcategorias.

- ARITMETICOS:** Canais gerados a partir de operações algébricas sobre canais analógicos (Canais aritméticos, em essência, são do mesmo tipo dos canais analógicos).
- FFT:** Canais gerados a partir de uma transformada discreta de Fourier utilizando o algoritmo FFT sobre canais analógicos.

Os canais derivados FFT são representados por gráficos de categoria na vertical. O eixo

das abscissas apresenta as harmônicas do sinal, enquanto o eixo das ordenadas apresenta a magnitude. Caso os dois canais tenham frequências fundamentais diferentes, um novo gráfico será criado para diferenciá-los, já que não podem dividir a multiplicidade das frequências harmônicas.

A figura 3 exemplifica os principais tipos de representação empregados pela ferramenta.

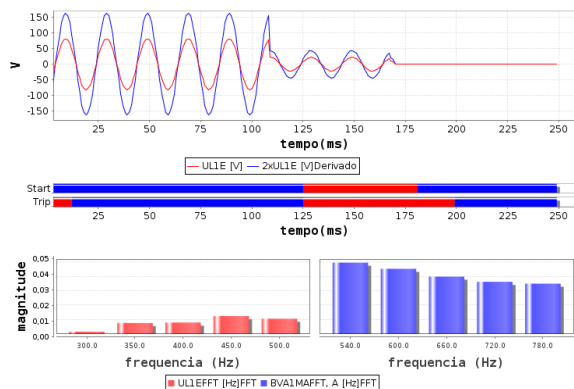


Figura 3: Representação de canais analógicos, derivados aritméticos, digitais e FFT

5 Diagrama Geral da Proposição

O fluxograma geral com representação esquemática dos processos realizáveis no sistema, é definido por três módulos interdependentes: Abertura e seleção de canais, *Math Builder* e Exibição de gráficos. A interação entre os módulos pode ser vista na figura 4. Cada módulo será melhor detalhado nas seções seguintes.

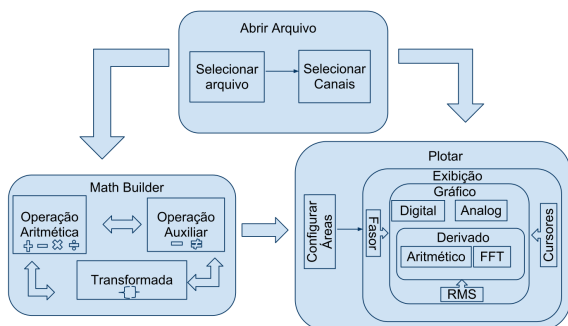


Figura 4: Fluxograma geral da proposição

5.1 Módulo 1: Abertura e seleção de canais

O conceito de “multi-arquivos” foi aplicado a ferramenta, permitindo a manipulação de vários canais entre arquivos distintos de forma simultânea. A abertura de um arquivo não inviabiliza a visualização de canais previamente selecionados. Canais importados podem ser habilitados e desabilitados do sistema, dando maior flexibilidade e clareza nas

manipulações. A figura 5 mostra um exemplo de seleção de canais no momento da abertura de um arquivo.

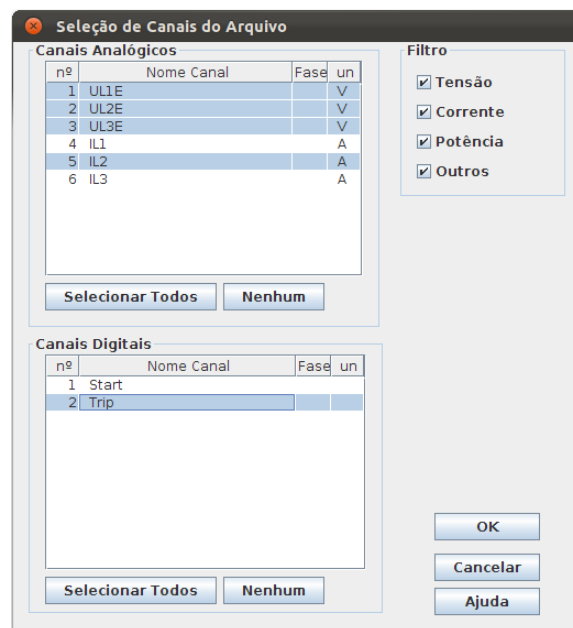


Figura 5: Seleção de canais na abertura de arquivo

5.2 Módulo 2: Math Builder

A ferramenta *Math Builder* provê manipulações de canais através da criação de novos objetos gerados a partir das operações matemáticas disponibilizadas. Tais objetos herdam as características do canal ao qual foi derivado podendo ser utilizado em novas operações ou na visualização de novos gráficos.

O *Math Builder* trabalha com operações apenas sobre canais analógicos e derivados. Não há operações sobre canais digitais. A figura 6 exibe uma tela típica de configuração e o escopo das operações é melhor detalhado nos itens a seguir.

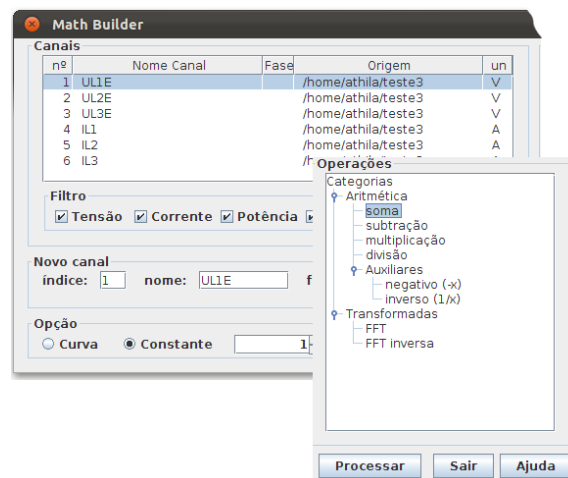


Figura 6: *Math Builder*

1. **Operações Aritméticas:** Operações Aritméticas são as operações mais básicas e só podem ser realizadas sobre canais analógicos. Dentre as operações disponíveis estão:

- (a) *soma*
- (b) *subtração*
- (c) *multiplicação*
- (d) *divisão*

2. **Operações Auxiliares:** As operações auxiliares dão recursos adicionais na manipulação dos sinais e inicialmente apenas duas operações estão disponíveis:

- (a) *negativo(-x)*
- (b) *inverso($\frac{1}{x}$)*

3. **Operações Transformadas:** Operações de transformadas utilizam um novo domínio de referência diferente do tempo. Este domínio, denominado *Domínio da Frequência*, representa seus sinais em componentes espectrais e por isso existem algumas restrições de operações conforme o tipo de canal. A seguir são listados os algoritmos implementados que transformam sinais analógicos discretos no tempo em sinais espectrais no domínio da frequência respectivamente.

- (a) *FFT*
- (b) *FFT inversa*

5.3 Modulo 3: Exibir Gráficos

Antes de exibir os dados importados é necessário realizar algumas configurações. Isso é necessário, pois o sistema deve tratar diversas peculiaridades como a aglutinação de canais oriundos de arquivos diferentes. Além disso, existem canais incompatíveis que não podem ser exibidos em uma mesma área devido à natureza de suas representações.

Cada área pode receber vários canais de diferentes arquivos, conforme pode ser visualizado na figura 7. Quando isso acontece os dados dos diferentes canais são mesclados em uma mesma área de exibição, ou seja, uma mesma área conterá curvas de diferentes canais. Existem restrições de exibição conjunta de tipos de canais. Com isso, o sistema não permite a exibição de gráficos, em uma mesma área, de canais cujos tipos são de naturezas distintas.

Outras formas de representação das curvas estão disponíveis conforme as seguintes opções:

1. Valor RMS

Cada canal analógico ou derivado aritmético terá um gráfico com o valor RMS correspondente. O valor RMS representa o valor quadrático eficaz da magnitude de uma quantidade variável. A figura 8 é um exemplo de

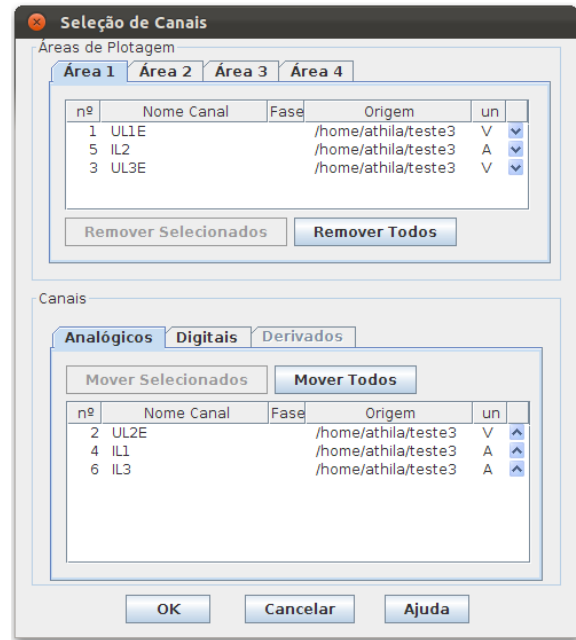


Figura 7: Seleção de canais para geração de gráficos

representação RMS de dois canais analógicos genéricos.

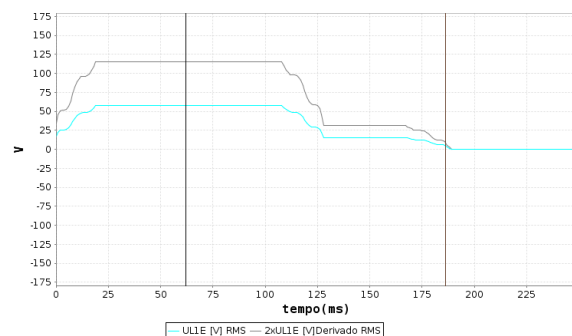


Figura 8: Representação RMS

2. Fasores

Cada canal analógico ou derivado aritmético terá uma série representando o ângulo e a amplitude máxima em um certo tempo. Essas séries serão mostradas em uma interface da própria janela interna de exibição de gráficos. Os controles deslizantes determinam o tempo que será representado o ângulo e a fase das séries nos gráficos.

Encontra-se nos painéis um gráfico do tipo vetorial, conforme mostrado na figura 9. Esse gráfico apresenta um vetor correspondente a cada série que é a projeção bidimensional do sinal no tempo. Na ponta desse vetor é mostrado sua amplitude e fase. Conforme o controle deslizante se movimenta o gráfico é atualizado.

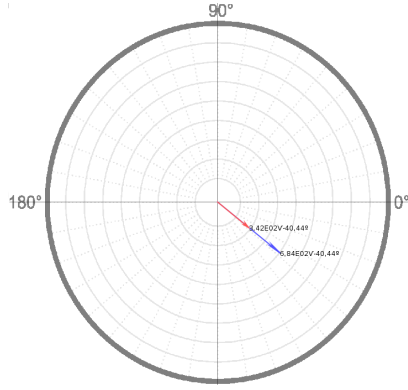


Figura 9: Representação fasorial

6 Testes Realizados

Testes foram realizados a partir de um projeto bem definido e que engloba a maioria das funcionalidades da ferramenta, tais como: a utilização do conceito de multi arquivos, uso de canais analógicos e digitais, geração de arquivos derivados aritméticos e de transformada, exibição de valores instantâneos e RMS, etc.

Para exemplificar e analisar a confiabilidade da ferramenta, o projeto de testes foi dividido em duas fases conforme explanados nas próximas seções.

6.1 Projeto de testes: Fase 1

O projeto da fase 1 pode ser visualizado na figura 10. Dois arquivos são selecionados. O primeiro arquivo possui tanto arquivos analógicos quanto digitais, mas foram selecionados apenas 4 canais analógicos (canais de tensão: *UL1E*, *UL2E* e *UL3E*; canais de corrente: *IL2* e *IL3*) e 2 canais digitais (*Start* e *Trip*) de seu repositório. O segundo arquivo possui apenas canais analógicos e serão selecionados 3 destes canais (*BVA1MA*, *BVA1MC* e *SUBA*). Em seguida, foram realizadas duas operações aritméticas: soma entre dois canais analógicos (canais *UL1E* e *UL2E*) e “negativação” da amostra (canal (*SUBA*)). Para verificação dos resultados obtidos são mostrados os gráficos representativos de cada canal.

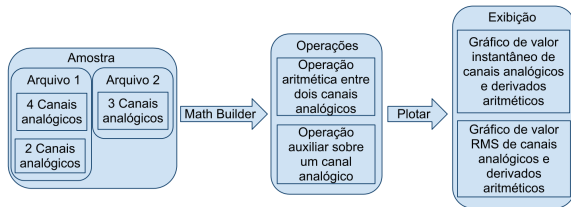


Figura 10: Planejamento de teste: Fase 1

A figura 11 mostra os valores instantâneos dos canais *UL1E*(vermelho), *UL2E*(azul) e do canal criado *UL1E+UL2E*(verde). Pode-se observar

que o canal *UL1E+UL2E* realmente representa a soma dos outros dois canais que estão defasados. Ele segue os canais ficando entre ambos. Perto do tempo de 110 ms, após o começo da amostragem, é possível ver a distorção do sinal *UL1E* ocorrida devida a uma falta elétrica. Como seu valor diminui em relação ao canal *UL2E*, o canal soma possui valores próximos a esse último canal.

Os canais *Start* e *Trip* são canais digitais e por isso possuem representações em formato de barras horizontais. Canais digitais geralmente representam eventos lógicos que ocorrem na linha de transmissão. Já a figura 12 mostra a relação entre o sinal *SUBA* e seu respectivo valor negativo *-SUBA*.

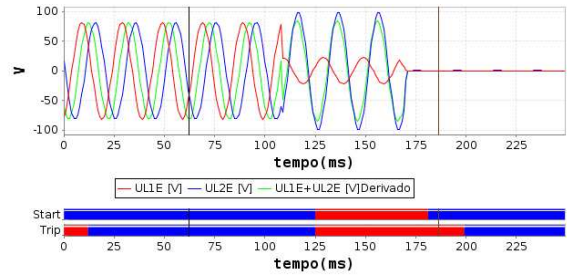


Figura 11: Fase 1: Canais analógicos, aritméticos e digitais

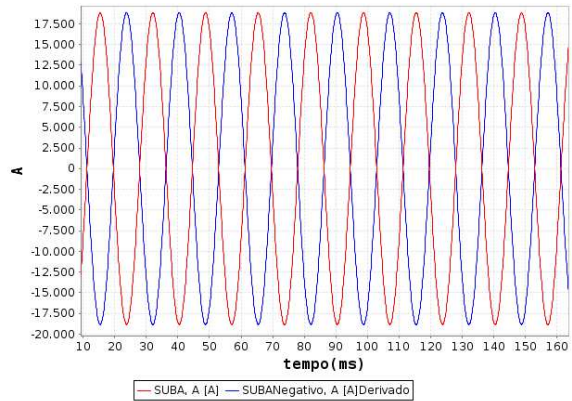


Figura 12: Fase 1: “Negativação” do canal analógico

A figura 13 é a representação dos canais em valores RMS. Pode-se perceber que o valor eficaz de todos os canais antes de ocorrer a falta se mantém constante e com mesmo valor. Isso ocorre porque os canais *UL1E* e *UL2E* estão defasados e como a soma entre ambos possui mesma amplitude o valor eficaz se mantém constante. Após a falta, o sinal *UL1E* cai em relação ao valor do canal *UL2E* e, portanto, o canal soma possui valor muito próximo do canal *UL2E*.

6.2 Projeto de testes: Fase 2

O projeto da fase 2 pode ser visualizado na figura 14. Nessa fase apenas um arquivo, contendo ape-

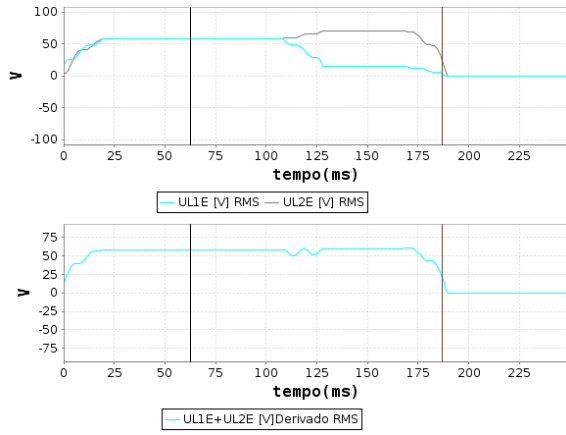


Figura 13: Fase 1: Valores RMS dos canais analógicos

nas canais analógicos, é selecionado. Aqui apenas um canal será selecionado (*BVA1MA*) para a aplicação da transformação rápida de Fourier. Para verificação dos resultados obtidos, os gráficos representativos do sinal original e transformado são apresentados.

A figura 15 aglutina os gráficos dos dois canais analisados. O primeiro gráfico diz respeito ao próprio sinal analógico *BVA1MA*. É uma representação no domínio do tempo. O segundo gráfico diz respeito a transformada FFT sobre esse canal. É uma representação no domínio da frequência e, portanto, apresenta o valor DC do sinal, bem como suas harmônicas nas frequências múltiplas da frequência fundamental.

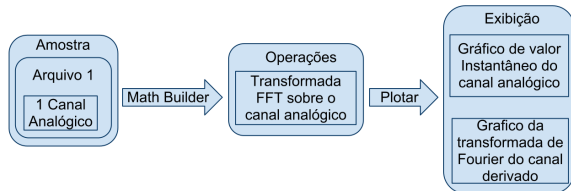


Figura 14: Planejamento de teste: Fase 2

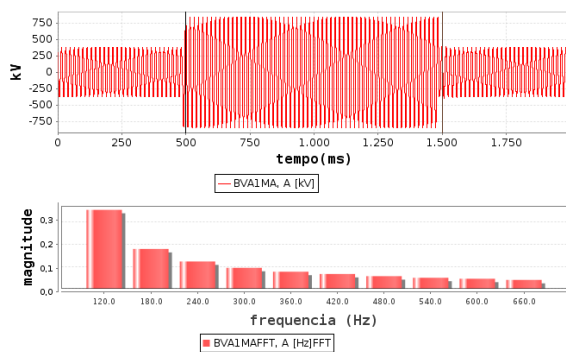


Figura 15: Fase 2: Valor instantâneo e representação espectral

Pode-se perceber que a visualização dos da-

dos é diferenciada para cada tipo de canal e, para a maioria dos canais, existe mais de um tipo de representação, ajudando o usuário da aplicação a obter o maior número de informações de forma rápida e fácil. Para aumentar a flexibilidade, a tela de configuração permite ao usuário definir como serão exibidas as diversas informações, podendo-se agrupá-las da forma requerida.

Canais são diferenciados por cor permitindo a distinção imediata de suas formas. As diversas legendas são separadas por gráfico e representam seus respectivos canais, indicando suas unidades de medidas que são expostas nos eixos coordenados.

Os resultados obtidos revelam a concordância com o valor esperado para as operações propostas, indicando a viabilidade da ferramenta. O teste revela a capacidade da ferramenta em um ambiente multi arquivos com canais distintos. A manipulação dos canais, com a criação de novos canais derivativos, ocorreu de forma simples, já que a interface foi desenvolvida para ser utilizada de forma amigável e intuitiva.

7 Conclusões

Este trabalho aborda a construção de uma nova ferramenta, denominada WAPS (*Wave Analyser for Power System*), voltada ao estudo e análise de perturbações e transitórios de redes de energia elétrica. Foram utilizados preceitos de *software* livre, permitindo à comunidade científica, voltada à área de proteção de sistemas elétricos de potência, utilizá-la de forma gratuita e adaptá-la as suas necessidades.

O uso dos conceitos de orientação a objetos permitiu construir um *software* modular e bem organizado, ajudando possíveis desenvolvedores que queiram modificá-lo. A utilização da linguagem Java permitiu a portabilidade essencial para disseminar a ferramenta em diversos ambientes com arquiteturas distintas.

O projeto de uma arquitetura modular permitiu também a construção de uma ferramenta independente do formato de entrada, a partir de uma estrutura comum a todos os módulos do *software*. Com isso, é possível implementar o reconhecimento de diversos formatos de entrada sem modificar o núcleo do código.

A ferramenta possui uma interface simples e amigável que permite ao usuário rápido aprendizado, além de fornecer um sistema de ajuda integrado em todas as telas do sistema. Mesmo a barra de status localizada na parte inferior exibe informações úteis que permite fácil navegação entre os módulos.

Cada tipo de canal possui representação própria, permitindo melhor visualização e extração de suas características intrínsecas. Assim, várias opções de representação estão disponíveis, tais como

a fasorial ou RMS. O módulo *Math Builder* permite a manipulação matemática de canais analógicos gerando novos canais também manipuláveis.

Desta forma, procurou-se desenvolver uma ferramenta que atendesse as necessidades do mercado, agregando ideias novas às funcionalidades já existentes, além de construir um produto final gratuito e flexível. Os autores esperam ter atingido os objetivos propostos e aberto novas oportunidades para extensão e continuidade do projeto.

Agradecimentos

Os autores agradecem o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo financiamento prestado, bem como ao Laboratório de Sistemas de Energia Elétrica (LSEE) da Escola de Engenharia Elétrica de São Carlos (EESC/USP) pela infra-estrutura fornecida durante o desenvolvimento deste.

Referências

- Agostini, M. N., Decker, I. C. and Silva, A. S. (2002). Desenvolvimento e implementação de uma base computacional orientada a objetos para aplicações em sistemas de energia elétrica, *SBA: Controle & Automação Sociedade Brasileira de Automação* **13**: 181–189.
- Barcellos, P., Oliveira, C. and Anido, M. (1996). A novel waveform analyzer for analog and digital signals for the windows(r) environment, *Circuits and Systems, 1996., IEEE 39th Midwest symposium on*, Vol. 1, pp. 385–388 vol.1.
- Branco, H. M. C., Monaro, R. M., Barbosa, D., Netto, U. C., Oleskovicz, M., Coury, D. V., Tokuno, S. E. and Parreira, O. R. M. (2010). Processo unificado aplicado ao projeto e desenvolvimento de um software compatível as análises de proteção e qualidade da energia elétrica em um sistema de distribuição real, *CBA: Congresso Brasileiro de Automação* **18**.
- Committee, I. S. C. (2004). IEEE recommended practice for the transfer of power quality data, *IEEE Std 1159.3-2003* pp. 1–119.
- Committee, P. (1991). IEEE standard common format for transient data exchange (comtrade) for power systems, *IEEE Std C37.111-1991* p. 1.
- Committee, P. (1999). IEEE standard common format for transient data exchange (comtrade) for power systems, *IEEE Std C37.111-1999* p. i.
- Dam, Q. and Meliopoulos, A. (2007). Relay simulation and testing software on the .net framework environment, *Power Engineering Society General Meeting, 2007. IEEE*, pp. 1–5.
- Kiessling, G. and Schwabe, S. (2004). Software solution for fault record analysis in power transmission and distribution, *Developments in Power System Protection, 2004. Eighth IEE International Conference on*, Vol. 1, pp. 192–195 Vol.1.
- King, J. and Gunther, E. (2006). COMTRADE / PQDIF conversion, *Transmission and Distribution Conference and Exhibition, 2005/2006 IEEE PES*, pp. 359–364.
- Larman, C. (2002). *Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos*, Bookman.
- Losi, A. and Russo, M. (2005). Dispersed generation modeling for object-oriented distribution load flow, *Power Delivery, IEEE Transactions on* **20**(2): 1532–1540.
- McDermott, T. (2006). An open source distribution system simulator, *Power Engineering Society General Meeting, 2006. IEEE*, p. 4 pp.
- Peharda, D., Hebel, Z. and Barta, A. (2007). Power system topology assessment and pre-estimation in an object oriented environment, *AFRICON 2007*, pp. 1–7.
- Poo, D., Kiong, D. and Ashok, S. (2007). *Object-Oriented Programming and Java*, Springer.
- Qiu, X. and Wimmer, W. (2000). Applying object-orientation and component technology to architecture design of power system monitoring, *Power System Technology, 2000. Proceedings. PowerCon 2000. International Conference on*, Vol. 2, pp. 589–594 vol.2.