



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

RENATO WESELY ALVES SOUTO

**APLICATIVO PARA MONITORAMENTO DAS MEDIÇÕES DAS
CHAVES SECCIONADORAS E RELIGADORES
AUTOMATIZADOS DA ENERGISA**

Campina Grande, Paraíba
Dezembro de 2010

RENATO WESELY ALVES SOUTO

APLICATIVO PARA MONITORAMENTO DAS MEDIÇÕES DAS
CHAVES SECCIONADORAS E RELIGADORES
AUTOMATIZADOS DA ENERGISA

*Trabalho de Conclusão de Curso submetido à
Coordenação do Curso de Graduação em
Engenharia Elétrica da Universidade Federal
de Campina Grande como parte dos requisitos
necessários para a obtenção do grau de
Bacharel em Ciências no Domínio da
Engenharia Elétrica.*

Área de Concentração: Eletrotécnica

Orientador:

Professor George Rossany Soares de Lira

Campina Grande, Paraíba
Dezembro de 2010

RENATO WESELY ALVES SOUTO

APLICATIVO PARA MONITORAMENTO DAS MEDIÇÕES DAS
CHAVES SECCIONADORAS E RELIGADORES
AUTOMATIZADOS DA ENERGISA

*Trabalho de Conclusão de Curso submetido à
Coordenação do Curso de Graduação em
Engenharia Elétrica da Universidade Federal
de Campina Grande como parte dos requisitos
necessários para a obtenção do grau de
Bacharel em Ciências no Domínio da
Engenharia Elétrica.*

Área de Concentração: Eletrotécnica

Aprovado em 10 / 12 / 2010

Professor Avaliador

Universidade Federal de Campina Grande
Avaliador

Professor George Rossany Soares de Lira

Universidade Federal de Campina Grande
Orientador, UFCG

Dedico este trabalho à minha mãe, que sempre me apoiou nos momentos mais difíceis e que sempre acreditou na minha capacidade profissional.

AGRADECIMENTOS

Primeiramente a Deus, que me proporcionou todas as ferramentas necessárias para que eu pudesse alcançar os meus objetivos.

A minha família, principalmente a minha mãe, Carminha, que sempre priorizou pela minha educação e foi quem mais me incentivou para que eu tivesse uma boa formação profissional. A meus irmãos Demerval, Bruno e Ana Cláudia, que sempre me deram suporte nos momentos mais difíceis.

As minhas tias, em especial a Tia Rosa, Dorinha e Corrinha, que me deram apoio em momentos importantes na minha vida.

Ao meu orientador de estágio, George Rossany, que me guiou na realização deste estágio.

Ao grande amigo Wellinsílvio, que sempre esteve disposto a me ajudar quando necessitei.

Ao engenheiro Glêston Agra, pelos conhecimentos transmitidos.

A todos que acreditaram na minha capacidade de tornar-me um bom engenheiro eletricista.

RESUMO

Neste relatório de trabalho de conclusão de curso, é relatado o desenvolvimento e funcionamento de um software aplicativo para monitoramento dos valores de grandezas elétricas, como correntes, tensões e potências, dos equipamentos automatizados da Energisa Paraíba, constituídos por chaves e religadores. Também é simulada algumas situações que são tipicamente demandadas por empresas do setor, como análise do fator de potência, desequilíbrio de cargas, fator de carga etc.

Palavras-chave: Software, Religadores, Chaves Seccionadoras, VTS.

SUMÁRIO

1	Introdução.....	1
2	Fundamentação Teórica.....	2
2.1	Índices de Qualidade.....	2
2.1.1	Qualidade do Produto.....	3
2.1.1.1	Tensão em Regime Permanente.....	3
2.1.1.2	Fator de Potência.....	4
2.1.1.3	Desequilíbrio de Tensão.....	5
2.1.1.4	Flutuação de Tensão.....	5
2.1.2	Qualidade do Produto.....	6
2.2	Análise de Carga.....	7
2.2.1	Fator de Carga.....	8
2.2.2	Fator de Demanda.....	8
2.2.3	Fator de Diversidade.....	9
2.2.4	Características das Cargas.....	9
2.3	Equipamentos Automatizados.....	10
2.3.1	Religadores.....	10
2.3.2	Chaves Automatizadas.....	15
2.4	Arquitetura de Comunicação.....	18
2.4.1	Modem GPRS.....	18
2.4.2	Rádio.....	20
2.5	VTS (Visual Tag System).....	21
2.5.1	Funcionamento.....	22
2.5.2	Tag.....	23
2.5.3	Alarmes.....	24
2.5.4	Confiabilidade.....	25
3	Aplicação.....	26
3.1	Ambiente de Desenvolvimento Aplicado.....	26
3.2	Funcionamento da Aplicação.....	27
4	Análise dos Resultados.....	32
4.1	Exemplo 1: Tensão em Regime Permanente.....	32
4.2	Exemplo 2: Fator de Potência.....	33
4.3	Exemplo 3: Análise de Indicador de Continuidade.....	34
4.4	Exemplo 4: Fator de Carga.....	35
4.5	Exemplo 5: Desequilíbrio de Fases.....	36
5	Conclusão.....	38
	Bibliografia.....	39
	APÊNDICE A - Código da Aplicação.....	40

1 INTRODUÇÃO

Um dos principais objetivos do setor da distribuição de energia é fornecer energia de boa qualidade aos consumidores, para isto, é necessário monitorar os equipamentos para melhorar os índices de qualidade do fornecimento de energia elétrica, estabelecidos pela ANEEL (Agência Nacional de Energia Elétrica).

Durante o estágio realizado na Energisa Paraíba, foi observada uma necessidade de um acompanhamento rápido e seguro das medições dos equipamentos automatizados da distribuição de energia, constituídos por chaves seccionadoras e religadores. Estas medições são basicamente relacionadas a tensão, corrente e potências ativa e reativa.

Então, diante desta necessidade, houve uma proposta de desenvolvimento de uma aplicação que facilitasse o monitoramento destes equipamentos, fornecendo as informações das medições dos mesmos em formato de tabelas e gráficos.

Estas informações têm como objetivo fornecer suporte para análise de curva de cargas, fatores de qualidade de serviços e produtos fornecidos, planejamento para redução de perdas, correção de desequilíbrio de fase etc.

Neste trabalho de conclusão de curso, será relatado alguns aspectos no âmbito de índice de qualidade de serviços e produtos relacionados à energia elétrica, assim como uma análise de carga.

Também, será descrito o funcionamento dos equipamentos automatizados monitorados, assim como os meios de comunicação utilizados. Em seguida, será realizada uma abordagem sobre o Supervisório utilizado pela Energisa Paraíba, o VTS, e por fim, será abordada toda estrutura da aplicação desenvolvida, desde sua concepção até os resultados obtidos.

2 FUNDAMENTAÇÃO TEÓRICA

O software aplicativo desenvolvido é voltado para o monitoramento das informações das medições dos equipamentos automatizados da Energisa Paraíba, compostos por chaves seccionadoras e religadores, respectivamente. Estes equipamentos, por sua vez, se comunicam através de rádio ou modem GPRS (General Packet Radio Service), com o Centro de Operação da Distribuição (COD) através de um sistema SCADA (Supervisory Control and Data Acquisition), que possui como software de interface, o VTS (Virtual Tag System)

Além do estudo a respeito da aquisição e tratamento dos valores medidos, será realizada uma análise sobre os principais aspectos relacionados aos índices de qualidade de energia elétrica, sendo ressaltada a utilidade do software desenvolvido. Também será abordada uma análise de carga, citando seus principais fatores e características.

2.1 ÍNDICES DE QUALIDADE

Os Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST), em seu módulo 8, estabelece procedimentos relacionados a qualidade de energia elétrica, abordando a qualidade do produto e a qualidade do serviço prestado.

Com relação à qualidade do produto, o módulo define parâmetros relativos à tensão em regime permanente e suas perturbações. Já no que diz respeito à qualidade de serviços prestados, o módulo menciona indicadores de qualidade e de tempo de atendimento.

Através do software aplicativo, é possível visualizar alguns picos de tensão, através da função valor máximo, e verificar se estes valores estão dentro da margem estabelecida pela ANEEL. Também é possível analisar se houve interrupção no fornecimento de energia analisando os níveis de corrente e tensão.

2.1.1 QUALIDADE DO PRODUTO

São considerados os seguintes aspectos com relação à qualidade da energia elétrica: tensão em regime permanente, fator de potência, harmônicos, desequilíbrio de tensão, flutuação de tensão, variações de tensão de curta duração e variação de frequência. A seguir, serão definidos os principais aspectos.

2.1.1.1 TENSÃO EM REGIME PERMANENTE

O regime permanente se caracteriza pelo período de tempo da leitura de tensão, onde não ocorrem distúrbios elétricos capazes de invalidar a leitura, definido como sendo de 10 minutos. Na Figura 1, a tensão de atendimento é definida pelo incremento das faixas de leituras e torno da tensão de referência (T_R):

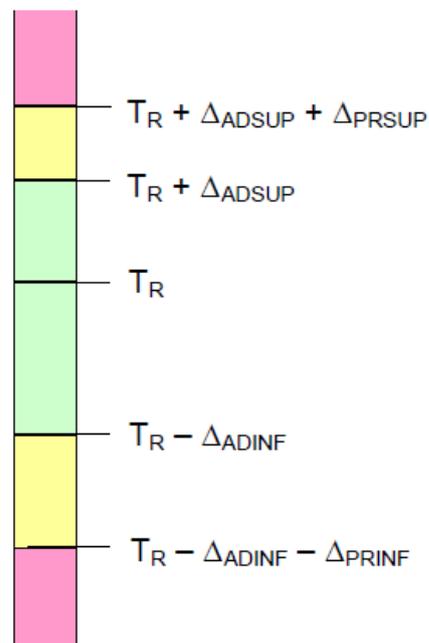


Figura 1: Faixas de tensão em relação à de referência

Onde:

- Tensão de referência (T_R);
- Faixa Adequada de Tensão ($T_R - \Delta_{ADINF}$, $T_R + \Delta_{ADSUP}$);
- Faixa Precária de Tensão ($T_R + \Delta_{ADSUP}$, $T_R + \Delta_{ADSUP} + \Delta_{PRSUP}$ ou $T_R - \Delta_{ADINF} - \Delta_{PRINF}$, $T_R - \Delta_{ADINF}$);
- Faixas Críticas de Tensão ($>T_R + \Delta_{ADSUP} + \Delta_{PRSUP}$ ou $< T_R - \Delta_{ADINF} - \Delta_{PRINF}$).

A seguir, estão apresentados nas Tabelas 1 e 2, os limites de cada faixa de tensão para os níveis de tensão de 13,8 kV e para 380/220 V, mais utilizados na rede de distribuição de energia:

Tabela 1: Limites de cada faixa de tensão para 13,8 kV

FAIXAS DE TENSÃO DE ATENDIMENTO PARA 13,8 KV	
Tensão de atendimento	Faixa de variação da tensão de leitura (TL) em relação à tensão de referência (TR)
Adequada	$0,93TR \leq TL \leq 1,05TR$
Precária	$0,90TR \leq TL < 0,93TR$
Crítica	$TL < 0,90TR$ ou $TL > 1,05TR$

Tabela 2: Limites de cada faixa de tensão para 380/220 V

FAIXAS DE TENSÃO DE ATENDIMENTO PARA 380/220 V	
Tensão de atendimento	Faixa de variação da tensão de leitura (Volts)
Adequada	$(348 \leq TL \leq 396)/(201 \leq TL \leq 231)$
Precária	$(327 \leq TL < 348$ ou $396 < TL \leq 403)/$ $(189 \leq TL < 201$ ou $231 < TL \leq 233)$
Crítica	$(TL < 327$ ou $TL > 403)/(TL < 189$ ou $TL > 233)$

2.1.1.2 FATOR DE POTÊNCIA

Fator de potência pode ser definido como a relação entre o componente ativo da potência e o valor total desta mesma potência. O fator de potência é um número adimensional. Este fator é calculado a partir de valores das potências ativa e reativa (P, Q) ou das respectivas energias (EA, ER), segundo os procedimentos PRODIST, usando-se as fórmulas a seguir:

$$fp = \frac{P}{\sqrt{P^2+Q^2}}, \quad (1)$$

ou

$$fp = \frac{E_A}{\sqrt{E_A^2+E_R^2}}. \quad (2)$$

De acordo com a regulação vigente, para unidades consumidoras atendidas com tensão inferior a 230 KV, o fator de potência deve estar entre 0,92 e 1 indutivo ou 1 e 0,92 capacitivo.

2.1.1.3 DESEQUILÍBRIO DE TENSÃO

O desequilíbrio de tensão ocorre quando há alterações dos valores padrões no sistema de distribuição. Segue as equações para cálculo de desequilíbrio:

$$FD\% = \frac{V_-}{V_+} \times 100, \quad (3)$$

ou

$$FD\% = 100 \sqrt{\frac{1 - \sqrt{3 - 6\beta}}{1 + \sqrt{3 - 6\beta}}}, \quad (4)$$

onde:

$$\beta = \frac{V_{ab}^4 + V_{bc}^4 + V_{ca}^4}{(V_{ab}^2 + V_{bc}^2 + V_{ca}^2)^2}. \quad (5)$$

O valor de referência nos barramentos do sistema de distribuição, segundo o PRODIST, deve ser igual ou inferior a 2%.

2.1.1.4 FLUTUAÇÃO DE TENSÃO

O objetivo da avaliação deste fenômeno dar-se á pelo efeito provocado pela cintilação luminosa em consumidores que tenham em sua unidade consumidora pontos de iluminação alimentados em baixa tensão. A flutuação de tensão é uma variação aleatória do valor eficaz da tensão.

A Severidade de Curta Duração (Pst) e Severidade de Longa Duração (Plt) são os indicadores para esse fenômeno. Segue o cálculo da Pst:

$$P_{st} = \sqrt{0,0314P_{0,1} + 0,0525P_1 + 0,0657P_3 + 0,28P_{10} + 0,08P_{50}}. \quad (6)$$

Onde: Pi (i = 0,1; 1; 3; 10; 50) corresponde ao nível de sensação de cintilação que foi ultrapassado durante i% do tempo. Na Figura 2, têm-se as faixas de tensão em relação à de referência.

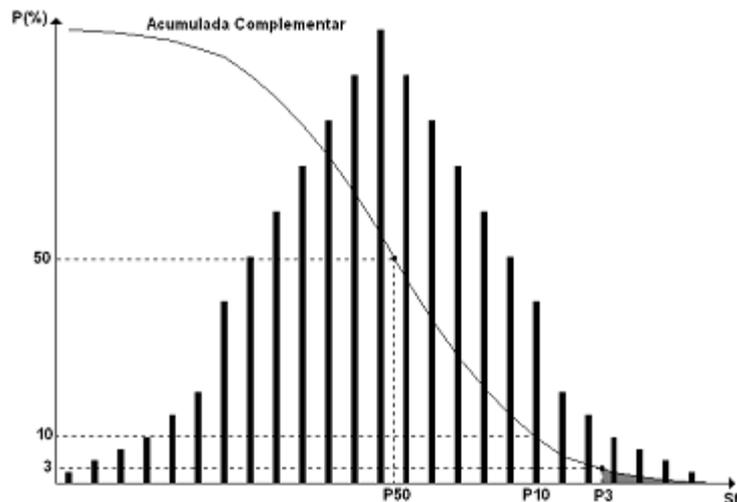


Figura 2: Faixas de tensão em relação à de referência

2.1.2 QUALIDADE DO PRODUTO

A criação de indicadores de qualidade do serviço prestado pelas distribuidoras tem como objetivo fornecer mecanismos para acompanhamento e controle do seu desempenho, oferecer aos consumidores parâmetros para avaliação dos serviços prestados pela distribuidora e fornecer subsídios para os planos de reforma, melhoramento e expansão da infra-estrutura das mesmas.

Os indicadores de qualidade do serviço classificam-se em indicadores de tempo de atendimento e em indicadores de continuidade. A seguir será detalhado o indicador relacionado à continuidade do serviço, que é um dos objetivos de análise deste software aplicativo.

Os indicadores de continuidade podem ser classificados quanto à duração e frequência da interrupção no fornecimento de energia elétrica e devem ser calculados e enviados à ANEEL mensalmente, trimestralmente e anualmente.

Entre os índices de continuidade coletivos estão o de Duração Equivalente de Interrupção por Unidade Consumidora (DEC) e o de Frequência Equivalente de Interrupção por Unidade Consumidora (FEC).

O DEC expressa a média do intervalo de tempo em que, no período de observação, ocorreu descontinuidade na distribuição de energia elétrica em cada unidade consumidora do conjunto considerado. É calculado de acordo com a seguinte fórmula:

$$DEC = \frac{\sum_{i=1}^k Ca(i) \times t(i)}{Cc} \quad (7)$$

onde: $Ca(i)$ é o número de unidades consumidoras interrompidas em um evento (i); $t(i)$ é a duração de cada evento (i), no período de apuração; Cc é o número total de unidades consumidoras, do conjunto considerado, no final do período de apuração.

O FEC expressa a média do número de interrupções ocorridas, no período de observação, em cada unidade consumidora do conjunto considerado. É calculado de acordo com a seguinte fórmula:

$$FEC = \frac{\sum_{i=1}^k Ca(i)}{Cc} \quad (8)$$

Os índices de continuidade individuais são o de Duração de Interrupção Individual por Unidade Consumidora (DIC), Frequência de Interrupção Individual por Unidade Consumidora (FIC) e Duração Máxima de Interrupção Contínua por Unidade Consumidora (DMIC).

O DIC é o intervalo de tempo em que, no período de observação, ocorreu descontinuidade na distribuição de energia elétrica em uma unidade consumidora ou ponto de conexão.

O FIC corresponde ao número de interrupções ocorridas, no período de observação, em cada unidade consumidora ou ponto de conexão.

O DMIC é o tempo máximo de interrupção contínua da energia elétrica em uma unidade consumidora ou ponto de conexão.

No estabelecimento dos limites de continuidade para os conjuntos de unidades consumidoras, é aplicada a técnica de análise comparativa de desempenho da distribuidora, tendo como referência os atributos físico-elétricos e dados históricos de DEC e FEC encaminhados à ANEEL.

2.2 ANÁLISE DE CARGA

A carga é um dos fatores mais importante de um sistema de distribuição, logo, é indispensável uma estimativa segura de suas características atual e de um futuro próximo, e conceito está diretamente relacionado à potência absorvida de uma fonte de energia. As cargas podem ser classificadas quanto a:

- Posição geográfica;

- Pagamento ou rentabilidade;
- Grau de dependência de suprimento;
- Tipo de atividade.

Com relação a posição geográfica, as cargas podem ser classificadas ainda em urbanas, suburbanas ou rurais. Já com relação ao faturamento, estas podem ser divididas em pequenos, médios e grandes consumidores. No que diz respeito ao tipo de atividade, as cargas se classificam em industrial, comercial, residencial e serviço público. As cargas consideradas especiais são as relacionadas a grandes hospitais e a algumas fábricas, pois são muito dependentes do suprimento de energia.

A representação gráfica em função do tempo é definida como curva de carga, que pode ser diária, semanal, mensal, anual ou qualquer outra, dependendo do estudo realizado. As curvas de cargas geralmente são diferentes a cada dia, e esta curva é intrínseca a cada consumidor e serve para definir o grupo em que o mesmo se enquadra.

2.2.1 FATOR DE CARGA

Com este índice, é possível verificar o quanto que a energia elétrica é utilizada de forma racional, e é dada pela razão entre a demanda média durante um período de tempo e a demanda máxima registrada neste período. O conceito de demanda é definido como o consumo de energia de uma instalação em relação ao tempo de consumo. A seguir, tem-se a fórmula correspondente ao fator de carga em função da demanda:

$$F_c = D_{\text{méd}} \text{ (kW ou kVA)} / D_{\text{máx}} \text{ (kW ou kVA)}, \quad (9)$$

onde: $D_{\text{méd}}$: Demanda média em um período de tempo, $D_{\text{máx}}$: Demanda máxima em um período de tempo.

2.2.2 FATOR DE DEMANDA

O fator de demanda de um sistema é dado pela relação entre demanda máxima e a potência total instalada na unidade consumidora. Com este fator, é possível analisar se os equipamentos estão sendo utilizados em plena carga.

2.2.3 FATOR DE DIVERSIDADE

Este fator informa a diversidade entre as demandas máximas de um grupo de consumidores, e é dada pela relação entre a soma das demandas máximas individuais e a demanda máxima do grupo.

2.2.4 CARACTERÍSTICAS DAS CARGAS

As cargas podem sendo tipo:

- Residencial: Constituída por alguns eletrodomésticos, lâmpadas etc. Possui fator de demanda de 70 à 100%, fator de diversidade 1,1 à 1,2 e fator de carga de 10 à 15%.
- Comercial: Possui mais cargas do que a residencial. O fator de demanda é entre 90 e 100%, fator de diversidade entre 1,1 e 1,2% e fator de carga entre 25 e 30%.
- Industrial: O fator de demanda de uma carga industrial pesada pode chegar a 90% e o fator de carga a 90%.
- Rural: Consiste basicamente de máquinas agrícolas e motores para bombas de irrigação. O fator de demanda está entre 90 e 100%, fator de diversidade entre 1 e 1,5% e fator de carga pode chegar a 30%.

Com a aplicação desenvolvida, é possível, observando a curva de carga, através do gráfico de potência, analisar o crescimento de carga de um dado consumidor.

Um aspecto importante no monitoramento de uma determinada carga, é com relação às perdas. Com um bom planejamento, é possível minimizar as perdas das seguintes formas: manutenção do nível de tensão utilizando reguladores de tensão; controle de fluxo de reativos, analisando bem as curvas de cargas e instalando banco de capacitores em paralelo em cargas indutivas; correção de desequilíbrio de fases, pois isto causa o aquecimento dos equipamentos, aumentando as perdas.

2.3 EQUIPAMENTOS AUTOMATIZADOS

Com relação aos equipamentos automatizados, chaves seccionadoras e religadores, foram estudados seus funcionamentos. A seguir, será relatado o estudo a respeito destes religadores e das chaves automatizadas.

2.3.1 RELIGADORES

Religadores são equipamentos de proteção muito utilizados em concessionárias de distribuição de energia, pois são capazes de diferenciar faltas permanentes de transitórias, sendo que estas últimas representam de 80 à 95% dos casos de falta ocorridos.

Este equipamento é capaz de realizar automaticamente comandos de abertura e fechamento de seus contatos quando ocorrer identificação de sobrecorrente no trecho que o mesmo se encontra alocado, e caso o religador identifique ainda um nível de sobrecorrente depois de alguns ciclos abertura e fechamento, o mesmo ficará com os contatos abertos em definitivo, isolando o trecho. A quantidade de ciclos, assim como os tempos de operação, ajustes de parâmetros de nível de corrente, tudo isto pode ser facilmente ajustados pelos usuários.

Os religadores da Energisa Paraíba, assim como os da Energisa Paraíba, são automatizados e controlados via remotamente. Os meios de comunicação utilizados são via modem GPRS, ou via rádio. As marcas dos religadores utilizados na EPB e na EBO são Nulec e Noja. Nas Figuras 3 e 4 estão apresentados esses tipos de religadores, respectivamente.



Figura 3: Religador Nulec

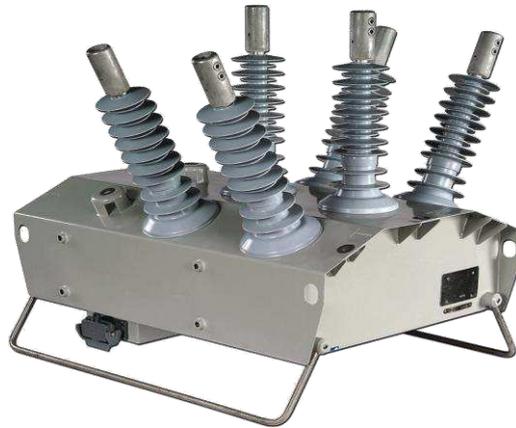


Figura 4: Religador Noja

Os principais componentes do religador são buchas de alta tensão, sinalizadores de estado atual (aberto ou fechado), contatos internos, sensores de alta tensão e corrente, mecanismos de extinção de arco e meio isolante.

O religador é responsável pela abertura física do circuito, tendo seu comando a partir do cubículo de controle. É no cubículo que ocorre a análise dos níveis de corrente e tensão, medidos pelos sensores do tanque principal, e é neste lugar onde é realizada a comunicação, via rádio ou modem GPRS, com o servidor.

A comunicação do religador com o cubículo é realizada por meio de um cabo de controle denominado cordão umbilical, que transfere as informações das medições dos sensores do tanque ao cubículo, e também viabiliza os comandos de abertura e fechamento estabelecidos pelo cubículo de controle ao tanque. É no cubículo onde ocorre a Ordem de Ajuste de Proteção (OAP), em que os parâmetros de proteção estabelecidos pela equipe de proteção de sistemas elétricos da Energisa Paraíba, do Departamento de Operação da Distribuição (DEOD), são inseridos no equipamento. Nas Figuras 5 e 6, a seguir, estão representadas os cubículos dos fabricantes Nulec e Noja, respectivamente.



Figura 5: Cubículo de controle Nulec



Figura 6: Cubículo de controle Noja

Este equipamento tem como função seccionar um circuito quando detectada uma sobrecorrente, e caso este sobrecorrente se extinga, o mesmo deve fechar o circuito novamente. Esta operação pode ser realizada algumas vezes, e caso a anomalia continue, o equipamento deverá seccionar de forma permanente, e não irá operar automaticamente até que ocorra uma intervenção humana.

O início do seu funcionamento ocorre a partir de um sensor que monitora a intensidade de corrente no circuito. Caso seja identificada uma sobrecorrente, é enviado um comando aos contatos do religador, e os mesmos se abrirão. Os contatos permanecerão neste estado durante o tempo de religamento, e depois fecharão seus contatos automaticamente.

Posteriormente, se o nível o sensor não identificar um nível de corrente elevado, o religador fechará seus contatos novamente, e depois do tempo de rearme, o mesmo poderá realizar o ciclo de religamento completo, que é composto por quatro aberturas e três religamentos.

Caso o sensor continue identificando um nível de sobrecorrente no circuito, então será realizado o mesmo processo, de abertura e fechamento, até os sensores não identificarem nenhuma falha ou até completar o ciclo de religamento. Quando isto ocorre, o religador permanecerá com seus contatos aberto, isolando o trecho em que o religador está operando.

Na Figura 7 têm-se os detalhes de um religador.

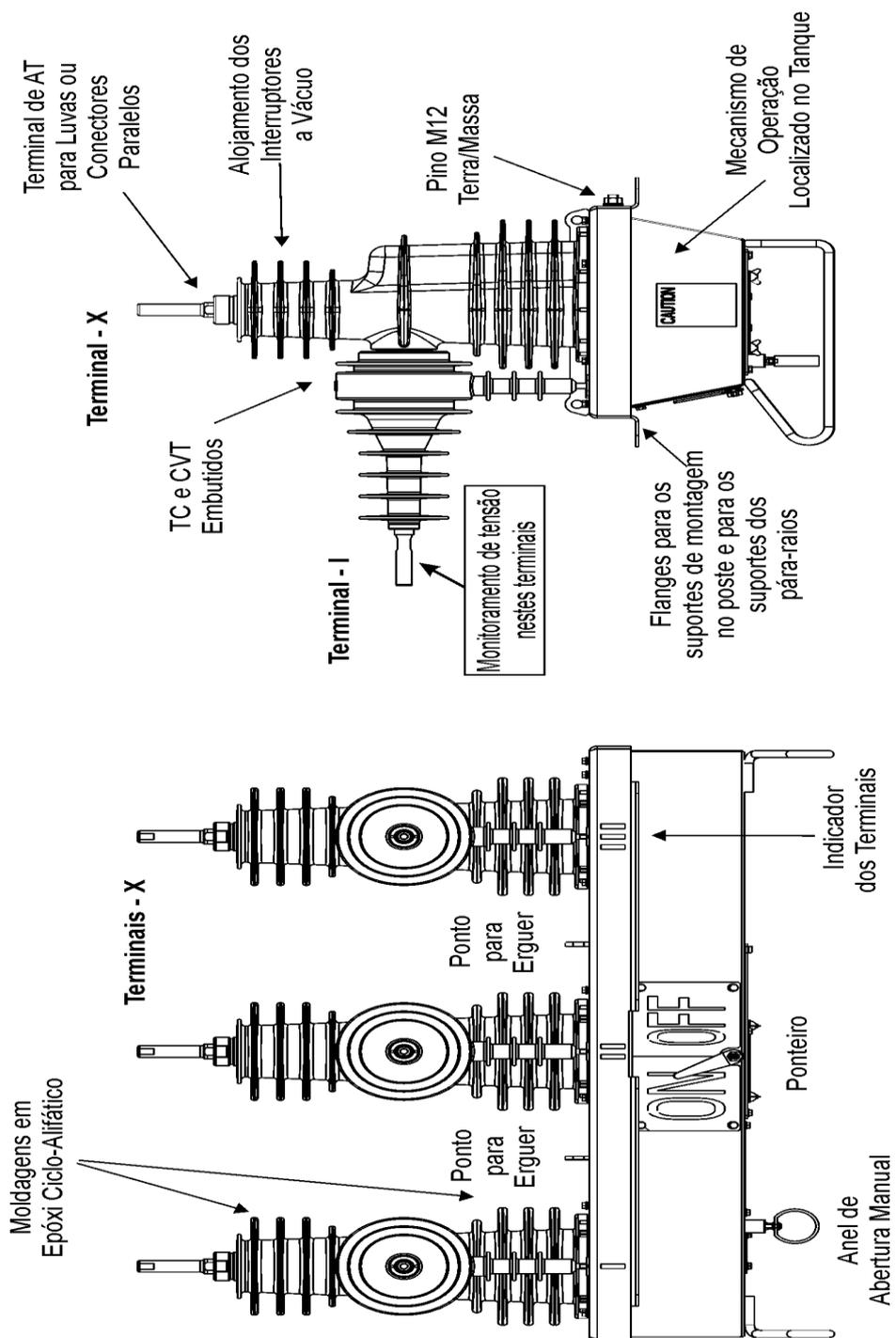


Figura 7: Recursos do Religador

2.3.2 CHAVES AUTOMATIZADAS

Estes equipamentos são muito semelhantes aos religadores. Quando as chaves automatizadas e os religadores estão em operação num alimentador em plena carga, estes equipamentos podem ser acionados normalmente. A diferença destes equipamentos está com relação a seus funcionamentos na presença de curto-circuito, pois as chaves automatizadas não podem abrir neste tipo de situação.

As chaves automatizadas podem ser usadas em interligação de alimentadores para realização de manobras ou até mesmo como alternativa para alimentação de uma determinada região.

As marcas das chaves automatizadas utilizadas pela EPB e EBO são da marca Nulec, a mesma utilizada nos religadores. Da mesma forma que ocorre nos religadores, estes equipamentos possuem um cubículo que realiza a análise das informações advindas dos sensores localizados no tanque principal.

A seguir, tem-se na Figura 8 a fotografia de uma chave automatizada.

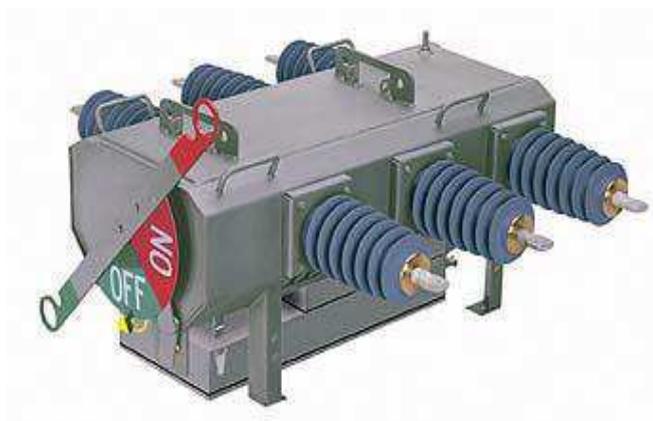


Figura 8: Tanque principal da chave automatizada

Quando o sensor, que está monitorando a corrente que passa no circuito que o religador está inserido, percebe um nível de sobrecorrente, as chaves ficam preparadas para iniciar o processo de contagem de ciclos do equipamento de retaguarda, neste caso o religador. As chaves não foram projetadas para atuar em correntes de curtos-circuitos, então, as mesmas atuam em coordenação com os religadores.

Se esta sobrecorrente for interrompida pelo equipamento de retaguarda, ou seja, pelo religador, então se inicia o processo de contagem de ciclos. Caso o valor de corrente a ser medido no próximo ciclo seja considerado normal, o religador fechará

seus contatos, e o contador de ciclos da chave será reiniciado. Caso contrário, a chave abrirá seus contatos, isolando o defeito.

A chave automatizada não possui meio de extinção de arco elétrico, logo, o mesmo só abrirá seus contatos quando o equipamento de retaguarda a montante, o religador, estiver na posição aberta. Se este equipamento não operasse em coordenação com o equipamento de retaguarda, o mesmo explodiria.

Estes equipamentos podem atuar como elementos de ligação entre alimentadores. Neste caso, as chaves automatizadas são normalmente abertas e são fechadas quando ocorrer a necessidade de suprimento de um alimentador através de outro alimentador.

Na Figura 9 têm-se os detalhes de uma chave seccionadora.

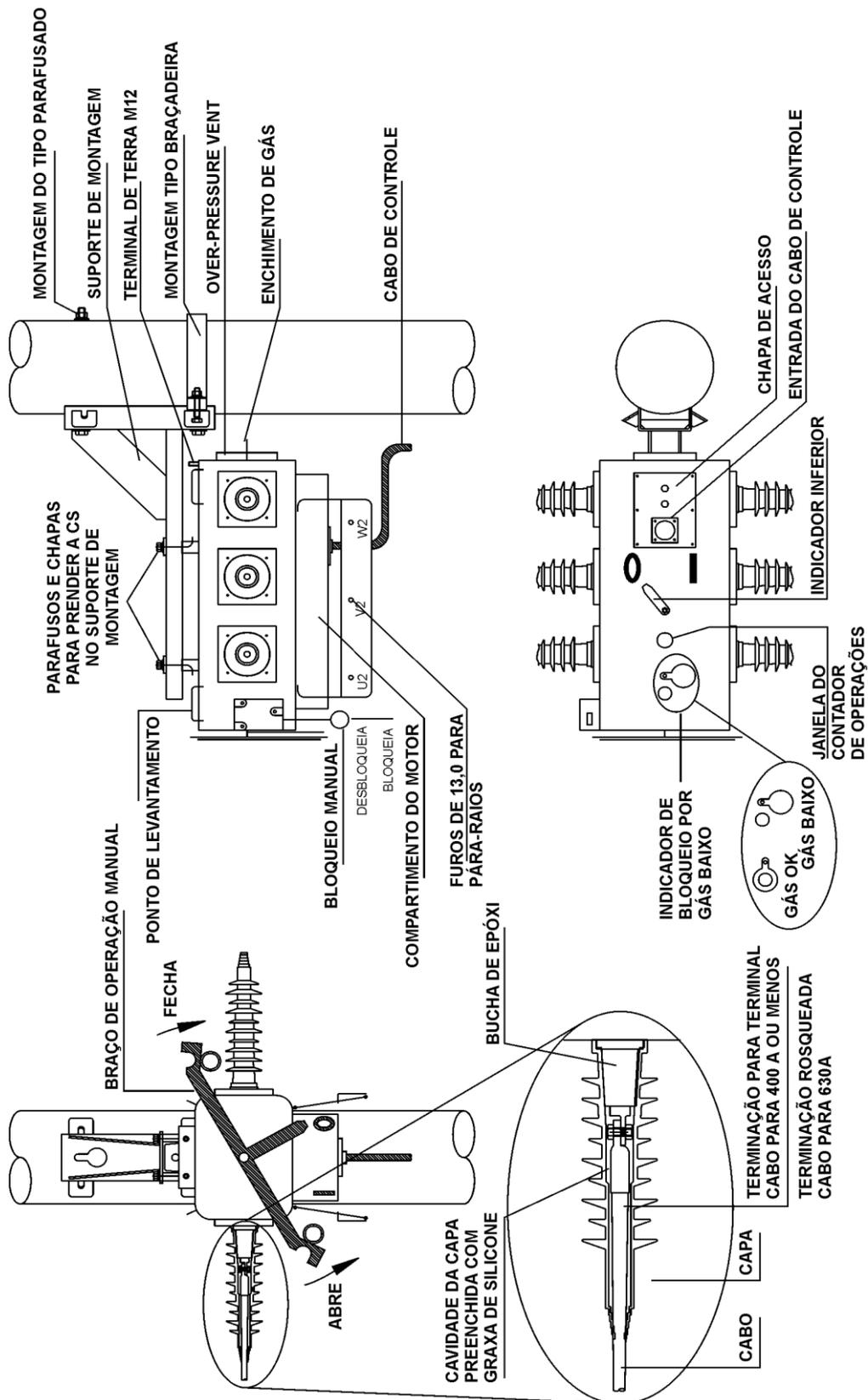


Figura 9: Detalhe da Chave Seccionadora

2.4 ARQUITETURA DE COMUNICAÇÃO

Os equipamentos automatizados da AD (Automação da Distribuição) se comunicam com o COD para transmitir dados das medições das grandezas elétricas e sobre o estado de funcionamento atual dos mesmos. Esta comunicação pode ser através de modem GPRS ou através de rádio analógico. O protocolo utilizado para comunicação entre o módulo de controle do cubículo e o Supervisório VTS é o DNP3 (Distributed Network Protocol).

2.4.1 MODEM GPRS

Dentre as duas opções que a EPB (Energisa Paraíba) e EBO (Energisa Borborema) possuem para comunicação dos equipamentos automatizados com o COD, rádio ou modem GPRS, estas têm preferência pelo modem GPRS em relação ao rádio por ter maior facilidade quanto à instalação, configuração e manutenção.

A arquitetura de comunicação via GPRS na EPB é dividida em quatro etapas: o conjunto de equipamentos de campo, a rede GPRS, os servidores da EPB e o COD, onde se encontra o Supervisório VTS. Na Figura 10, a seguir, tem-se a representação da arquitetura de comunicação via GPRS.

ARQUITETURA DO SISTEMA DE COMUNICAÇÃO GPRS DA ENERGISA-PB

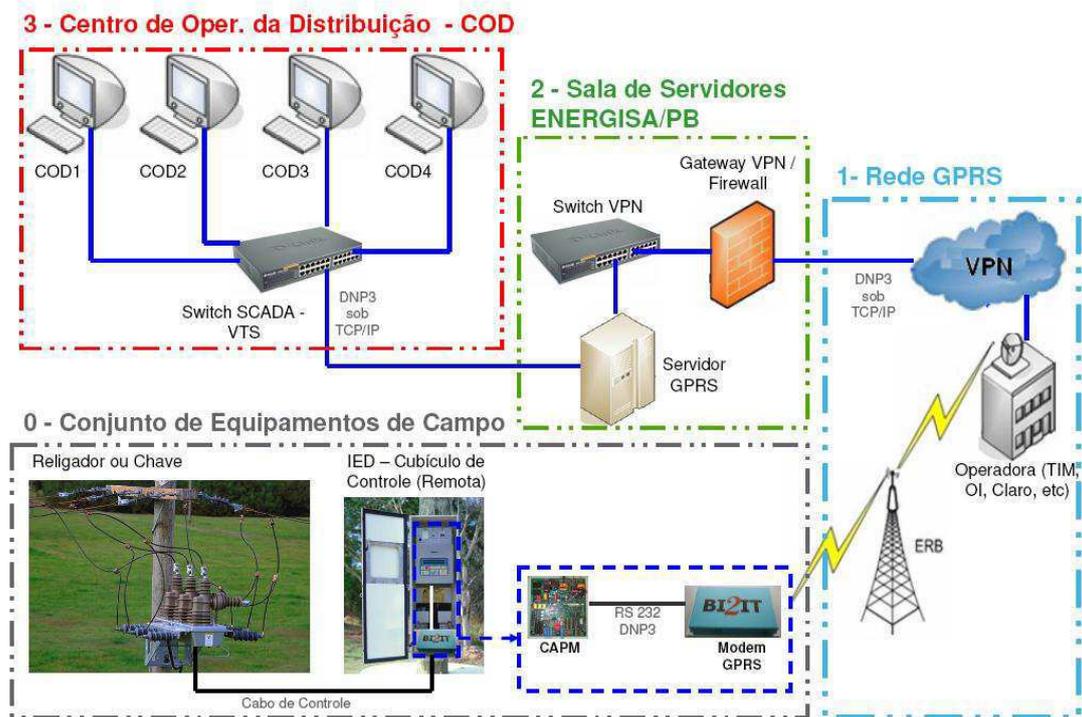


Figura 10: Arquitetura de comunicação via GPRS da EPB.

O modem GPRS fica instalado dentro do cubículo de controle do equipamento e se comunica com a placa de controle via protocolo DNP3. A comunicação realizada entre o equipamento e o servidor é feita através de operadoras disponíveis na região onde o equipamento se encontra. Para aumentar a confiabilidade da comunicação, o fabricante do modem disponibiliza a utilização de até dois chips, ou seja, caso a operadora preferencial perca a comunicação, a operadora redundante ficará responsável pela comunicação do equipamento. As informações transmitidas por essa rede são confidenciais, logo, para garantir maior segurança na transmissão desses dados, a operadora utiliza uma rede virtual privada (VPN).

As informações, então, são enviadas do GPRS ao servidor, mas antes passam por um firewall, com o objetivo de aumentar a segurança com relação aos dados transmitidos. Depois dos dados chegarem até o servidor, as informações são disponibilizadas ao COD, onde estão localizados os operadores que realizam o monitoramento do sistema de distribuição da EPB e EBO.

2.4.2 RÁDIO

A arquitetura a rádio é mais complexa em relação à arquitetura a modem GPRS. Mesmo esta arquitetura ser própria e utilizada em toda a empresa, as equipes da AD não possuem acesso a toda a estrutura da rede, o que limita o conhecimento dos técnicos com relação a esta tecnologia. Logo, os equipamentos que não possuem cobertura GPRS, utilizam o rádio para comunicação, e esta tecnologia é utilizada para comunicação de vários equipamentos da AD com o COD.

Esta arquitetura se divide em três níveis. O primeiro nível é composto por equipamentos como chaves automatizadas, religadores, relés, disjuntores, transformadores, enfim, equipamentos que compõem uma subestação (SE) ou uma rede de distribuição. Estes equipamentos se comunicam via rádio, de forma direta ou através de repetidoras (RPT), com as bases localizadas em subestações.

No segundo nível temos a unidade terminal remota (UTR), que capta e trata os dados enviados pelo rádio dos equipamentos. Cada subestação possui uma UTR e sua funcionalidade é de acordo com o setor responsável, podendo este ser da distribuição ou da subestação. Depois, com os dados adquiridos, a UTR, então, transmite as informações para um satélite através de uma antena.

No terceiro nível, o satélite recebe o sinal enviado pela UTR e envia para a sede da EPB. Neste nível, o sinal é tratado e colocado na rede SCADA, em que esta é uma abreviação de Sistemas de Supervisão e Aquisição de Dados. Então, a partir do VTS, com as informações advindas da rede SCADA, os operadores do COD terão acesso a informações dos equipamentos como sequência de eventos, alarmes emitidos, medições etc.

Na Figura 11, pode-se observar a arquitetura de comunicação da EPB e EBO:

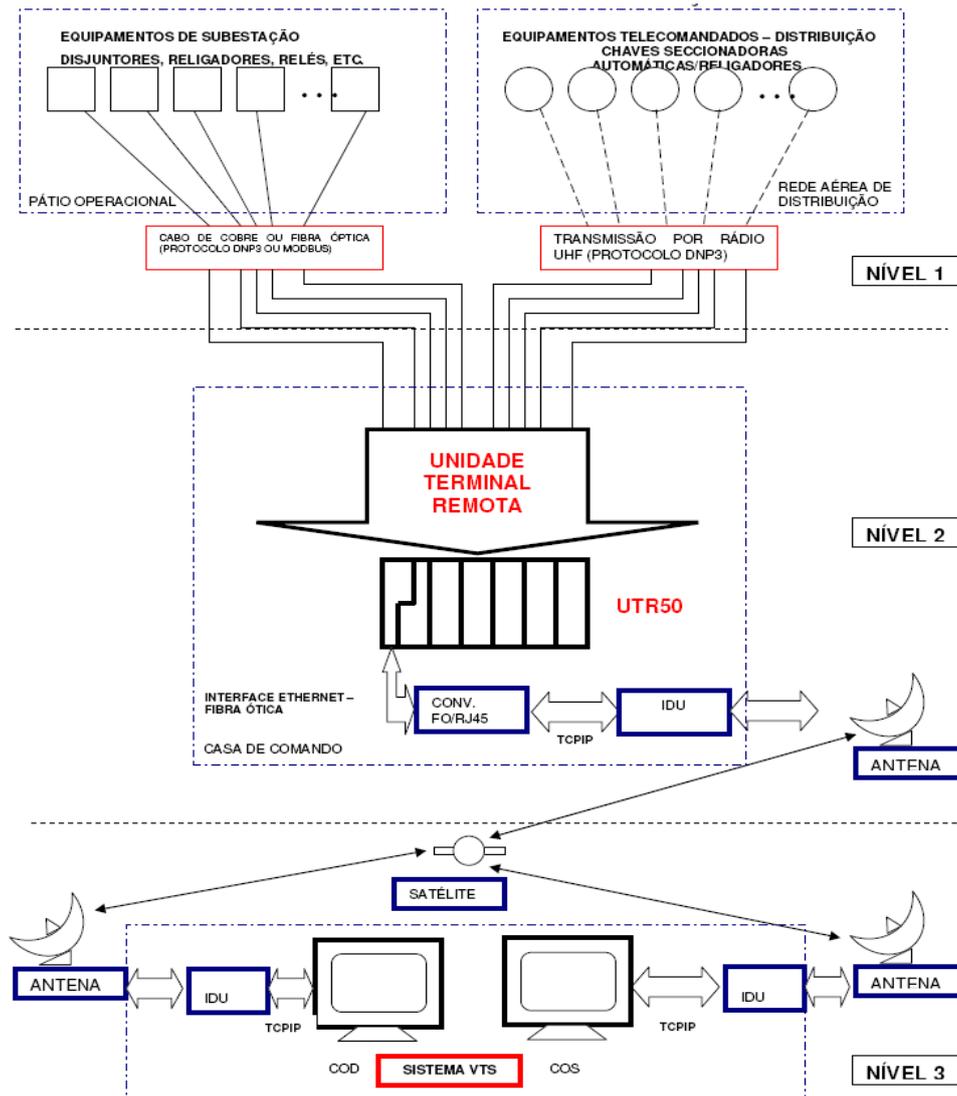


Figura 11: Arquitetura de comunicação via rádio

2.5 VTS (VISUAL TAG SYSTEM)

Para realização de monitoramento e supervisão dos equipamentos automatizados, é necessária a utilização de sistemas que utilizem softwares com interface gráfica para que os usuários realizem este monitoramento. Estes sistemas, utilizado pela Energisa Paraíba, são Sistemas de Supervisão e Aquisição de Dados, mais conhecidos como SCADA, e o software utilizado para realização desta interface gráfica do sistema SCADA é o VTS.

2.5.1 FUNCIONAMENTO

Este software de interface humana-máquina (IHM), VTS, desenvolvido pela Trihedral Engineering Limited, permite ao operador monitorar e controlar as ações sobre os equipamentos remotamente, através de um ou mais computadores. Na Energisa Paraíba, o COD realiza este monitoramento, possuindo sete computadores com a licença de uso do VTS.

O VTS habilita o controle de monitoramento remoto ao operador, através de uma conexão entre a Unidade Terminal Remota (UTR) e o dispositivo telecomandado. A UTR viabiliza a transmissão de dados entre os equipamentos e os computadores que utilizam o software VTS. Na Figura 12, tem-se uma analogia da comunicação entre o VTS e um dispositivo.

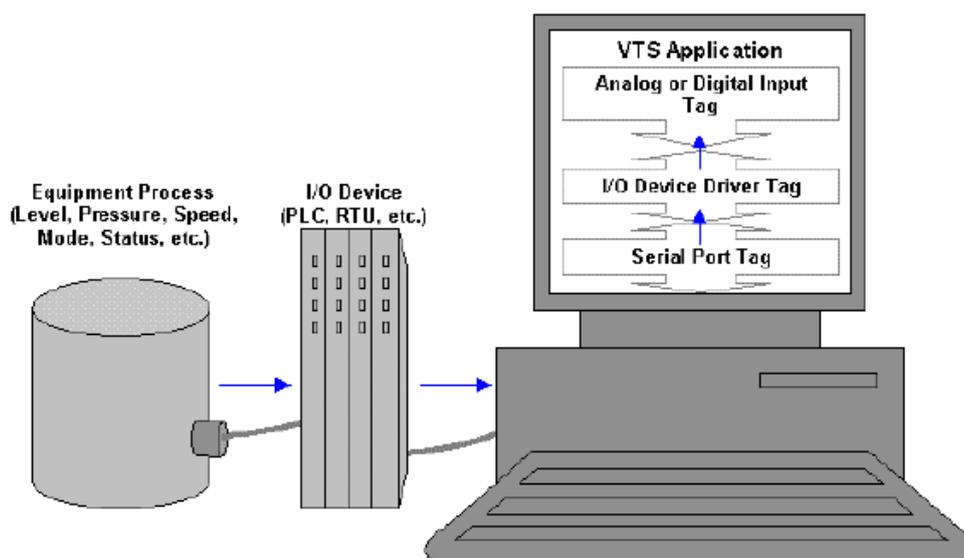


Figura 12: Analogia da comunicação entre o VTS e um dispositivo

Algumas das informações transmitidas são status dos equipamentos, como status de comunicação, os modos de operação, os valores de ajustes de proteção, valores das medições de corrente, tensão, potência ativa ou reativa etc. A seguir, na Figura 13, é possível visualizar os status de um grupo de equipamentos:

Tranche	Componente	Situação atual	Alimentador/SE	Comunicação	Good	Bad	Reset de Estatíst. Con	N Comun.	Controle de Serviço	Observação
LPT 1	65117	Fechado	01L3-CRH	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	RADIO
LPT 1	86623	Fechado	01L3-TRR	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	GPRS - LUFA
LPT 1	57329	Fechado	01L1-0TF	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	RADIO
LPT 1	57484	Fechado	01L2-0TF	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	RADIO
LPT 1	55700	Fechado	01L3-0TF	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	RADIO
LPT 1	65116	Fechado	01L3-ITO	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	RADIO
LPT 1	65232	Invalído	01L1-SPK	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	RADIO
LPT 2	82190	Invalído	01L5-S2R	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Local - NF	RADIO
LPT 2	82181	Fechado	01L5-SBT	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NA	RADIO
LPT 2	82187	Fechado	01L5-S2R	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Local - NF	RADIO
LPT 2	41924	Fechado	01L6-PTS	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	RADIO
LPT 2	59981	Fechado	01L2-S2L	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NA	RADIO
LPT 2	68583	Invalído	01L1-S2L	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	RADIO
LPT 2	82197	Invalído	01L2-JTB	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Local - NF	RADIO
LPT 3	83460	Fechado	01L4-BJC	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	RADIO
LPT 3	83474	Fechado	01L2-PTS	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	GPRS - LUFA
LPT 3	83475	Fechado	01L6-PTS	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	GPRS - LUFA
LPT 3	83480	Fechado	01L3-S2R	<input checked="" type="checkbox"/>	Show Comm			Show Stats	Operação Remota - NF	RADIO

Figura 13: Status dos equipamentos no VTS

Na Figura 13, é possível observar os status dos equipamentos, ou seja, se os mesmos estão abertos, fechados ou sem comunicação. Também é possível observar em qual alimentador os equipamentos estão alocados e ainda qual o tipo de comunicação que os mesmos utilizam, rádio ou modem GPRS.

2.5.2 TAG

Para que o VTS receba e envie dados ao equipamento, a aplicação é configurada para viabilizar cada processo do equipamento no sistema. Quando um processo é relacionado a uma única função do equipamento físico, como o valor de tensão de um religador, por exemplo, esta representação de relação é denominada de Tag.

As Tags são componentes de software que identificam, definem e representam cada processo do equipamento no sistema de controle. No desenvolvimento da aplicação neste trabalho de conclusão de curso, foram analisadas todas as Tags disponíveis no VTS. Através das Tags, foi possível armazenar informações no banco de dados Access através de uma configuração no próprio VTS.

A Tabela 3, a seguir, é um exemplo de várias Tags relacionadas a um determinado equipamento.

Tabela 3: Tags relacionadas ao religador ABR_RL_6651

Tag	Descrição
ABR_RL_6651_FAULT_IA	RL_6651 Corrente de Falta Fase A
ABR_RL_6651_FAULT_IB	RL_6651 Corrente de Falta Fase B
ABR_RL_6651_FAULT_IC	RL_6651 Corrente de Falta Fase C
ABR_RL_6651_FAULT_N	RL_6651 Corrente de Falta Neutro
ABR_RL_6651_FPOT	RL_6651 Fator de Potência
ABR_RL_6651_FREQ	RL_6651 Frequência
ABR_RL_6651_I_BAT	RL_6651 Corrente de Bateria
ABR_RL_6651_IA	RL_6651 Corrente Fase A
ABR_RL_6651_IB	RL_6651 Corrente Fase B
ABR_RL_6651_IC	RL_6651 Corrente Fase C
ABR_RL_6651_IN	RL_6651 Corrente Neutro
ABR_RL_6651_KVAR	RL_6651 Potencia Reativa
ABR_RL_6651_N_SERIE	RL_6651 Numero de Serie
ABR_RL_6651_OPER	RL_6651 Contador de Operação
ABR_RL_6651_RES_BAT	RL_6651 Capacidade Residual de Bateria
ABR_RL_6651_ST_MECANISMO	RL_6651 Contador de Status do Mecanismo
ABR_RL_6651_VAB	RL_6651 Tensão Fase A
ABR_RL_6651_VBC	RL_6651 Tensão Fase B
ABR_RL_6651_VCA	RL_6651 Tensão Fase C
ABR_RL_6651_VOLT_BAT	RL_6651 Tensão de Bateria
ABR_RL_6651_WATT	RL_6651 Potencia Ativa

2.5.3 ALARMES

O VTS pode ser configurado para gerar alarmes, informando ao operador quando ocorrer algo que não esteja previsto. No VTS, podem existir vários tipos de alarmes, como os referentes a equipamentos sem comunicação ou a corrente de falta, por exemplo. Na Figura 14, têm-se vários alarmes gerados pelo VTS da Energisa Paraíba.

The screenshot shows the 'Alarmes' window in the Energisa SCADA system. The interface includes a header with the Energisa logo, a title bar, and a menu bar. Below the menu bar, there are several control panels for filtering and displaying the alarm data. The main area is a table listing 153 alarms.

Evento	Prioridade	Data - Tempo	Área	Mensagem	Valor	Operador
Desativado	Urgencia	19 Ago - 07:50:19	OESTE	RL_83468 - Desbalanceamento de Cargas Alto	9.0 A	Logged Off
Recon	Advertencia	19 Ago - 07:49:39	OESTE	BTF_2112 - Relig. Linha_L2 BSF - Falha Comunicação		Alarme Anderson
Ativo	Advertencia	19 Ago - 07:48:39	OESTE	BTF_2112 - Relig. Linha_L2 BSF - Falha Comunicação		Alarme Logged Off
Desativado	Urgencia	19 Ago - 07:48:02	OESTE	RL_83469 - Desbalanceamento de Cargas Alto	9.0 A	Logged Off
Recon	Urgencia	19 Ago - 07:47:39	OESTE	RL_83469 - Desbalanceamento de Cargas Alto	10.0 A	douglas
Ativo	Urgencia	19 Ago - 07:47:36	OESTE	RL_83469 - Desbalanceamento de Cargas Alto	10.0 A	Logged Off
Desativado	Advertencia	19 Ago - 07:36:48	OESTE	RL_83476 - Equipamento sem Comunicação		Normal Logged Off
Recon	Urgencia	19 Ago - 07:36:09	OESTE	PBL_RL_89651 - Corrente de Falta Fase A	178.0 A	SOLONILTON
Recon	Urgencia	19 Ago - 07:36:09	OESTE	PBL_RL_89651 - Corrente de Falta Fase B	176.0 A	SOLONILTON
Ativo	Urgencia	19 Ago - 07:36:04	OESTE	PBL_RL_89651 - Corrente de Falta Fase A	178.0 A	Logged Off
Ativo	Urgencia	19 Ago - 07:36:04	OESTE	PBL_RL_89651 - Corrente de Falta Fase B	176.0 A	Logged Off
Recon	Advertencia	19 Ago - 07:34:27	OESTE	RL_83476 - Equipamento sem Comunicação		Alarme SOLONILTON
Ativo	Advertencia	19 Ago - 07:34:21	OESTE	RL_83476 - Equipamento sem Comunicação		Alarme Logged Off
Desativado	Advertencia	19 Ago - 07:15:38	OESTE	BTF_2111 - Falha Comunicação		Normal Logged Off
Recon	Emergencia	19 Ago - 07:13:49	OESTE	CH_83385 - Bucha A2		Com TensSOLONILTON
Recon	Emergencia	19 Ago - 07:13:49	OESTE	CH_83385 - Bucha B2		Com TensSOLONILTON
Recon	Emergencia	19 Ago - 07:13:49	OESTE	CH_83385 - Bucha C2		Com TensSOLONILTON
Recon	Urgencia	19 Ago - 07:13:49	OESTE	RL_88633 - Pickup de Corrente do Neutro	116.0 A	SOLONILTON
Recon	Emergencia	19 Ago - 07:13:49	OESTE	RL_88633 - Equipamento		Fechado SOLONILTON
Recon	Emergencia	19 Ago - 07:13:49	OESTE	RL_88633 - Pickup de Neutro Ultrapassado		Alarme SOLONILTON
Recon	Advertencia	19 Ago - 07:13:49	OESTE	BTF_2111 - Falha Comunicação		Alarme SOLONILTON
Recon	Emergencia	19 Ago - 07:13:49	OESTE	CH_83385 - Bucha A2		Com TensSOLONILTON
Recon	Emergencia	19 Ago - 07:13:49	OESTE	CH_83385 - Bucha B2		Com TensSOLONILTON
Recon	Emergencia	19 Ago - 07:13:49	OESTE	CH_83385 - Bucha C2		Com TensSOLONILTON
Recon	Urgencia	19 Ago - 07:13:49	OESTE	RL_88633 - Pickup de Corrente do Neutro	116.0 A	SOLONILTON
Recon	Emergencia	19 Ago - 07:13:49	OESTE	RL_88633 - Equipamento		Fechado SOLONILTON
Recon	Emergencia	19 Ago - 07:13:49	OESTE	RL_88633 - Pickup de Neutro Ultrapassado		Alarme SOLONILTON
Recon	Advertencia	19 Ago - 07:13:49	OESTE	BTF_2111 - Falha Comunicação		Alarme SOLONILTON
Desativado	Emergencia	19 Ago - 07:13:42	OESTE	RL_88633 - Equipamento		Fechado Logged Off
Desativado	Emergencia	19 Ago - 07:13:41	OESTE	CH_83385 - Bucha A2		Com TensLogged Off
Desativado	Emergencia	19 Ago - 07:13:41	OESTE	CH_83385 - Bucha C2		Com TensLogged Off
Desativado	Emergencia	19 Ago - 07:13:41	OESTE	CH_83385 - Bucha B2		Com TensLogged Off
Ativo	Emergencia	19 Ago - 07:13:39	OESTE	CH_83385 - Bucha B2		Sem TensLogged Off
Ativo	Emergencia	19 Ago - 07:13:39	OESTE	CH_83385 - Bucha A2		Sem TensLogged Off
Ativo	Emergencia	19 Ago - 07:13:39	OESTE	CH_83385 - Bucha C2		Sem TensLogged Off
Ativo	Urgencia	19 Ago - 07:13:37	OESTE	RL_88633 - Pickup de Corrente do Neutro	116.0 A	Logged Off
Ativo	Emergencia	19 Ago - 07:13:37	OESTE	RL_88633 - Pickup de Neutro Ultrapassado		Alarme Logged Off
Ativo	Emergencia	19 Ago - 07:13:37	OESTE	RL_88633 - Equipamento		Aberto Logged Off
Ativo	Advertencia	19 Ago - 07:13:30	OESTE	BTF_2111 - Falha Comunicação		Alarme Logged Off
Recon	Urgencia	19 Ago - 07:12:06	OESTE	RL_83468 - Desbalanceamento de Cargas Alto	10.0 A	douglas

Figura 14: Alarmes gerados pelo VTS

2.5.4 CONFIABILIDADE

No sistema SCADA da Energisa, existem dois servidores que são utilizados exclusivamente para o VTS, um chamado primário e o outro chamado secundário ou de backup. Esses servidores possuem base de dados idêntica, pois em caso de falha do servidor primário, o secundário passará a ser utilizado imediatamente. Quando o servidor primário volta a funcionar normalmente, se restabelece a condição inicial.

3 APLICAÇÃO

O software aplicativo foi desenvolvido para facilitar o monitoramento das medições dos equipamentos automatizados, num determinado período de tempo.

A Energisa utiliza o sistema operacional Windows e para o sistema de gerenciamento de banco de dados Microsoft Office Access, logo, decidiu-se por utilizar a linguagem C# para desenvolvimento do software aplicativo, uma vez que esta linguagem também constitui de uma ferramenta da Microsoft.

3.1 AMBIENTE DE DESENVOLVIMENTO APLICADO

C# (C Sharp) é uma linguagem de programação desenvolvida pela Microsoft. Com esta linguagem associada ao .NET Framework (plataforma de desenvolvimento da Microsoft), é possível o desenvolvimento de aplicativos para Windows, Webservices, ferramentas de banco de dados etc.

O ambiente de desenvolvimento da aplicação foi o Microsoft Visual Studio C# 2010 Express, que pode ser realizado o download de forma gratuita no seguinte site: <http://www.microsoft.com/express/Downloads/#2010-Visual-CS>.

O desenvolvimento da aplicação iniciou-se na configuração do VTS, para armazenar as informações das medições dos equipamentos automatizados em um banco de dados. O arquivo de configuração do VTS, localizado no diretório onde o VTS foi instalado, é o “REPORT.INI”. Neste arquivo, foram implementadas todas as Tags de interesse, depois de um prévio estudo sobre as mesmas.

Posteriormente, o VTS criou um arquivo de banco de dados denominado “relatórios.mdb”, com todas as informações das medições, definidas a partir das Tags. Estas informações foram armazenadas a cada quinze minutos, com o intuito de não sobrecarregar o sistema SCADA. Criado o banco de dados, então, iniciou-se o processo de desenvolvimento do software aplicativo propriamente dito.

3.2 FUNCIONAMENTO DA APLICAÇÃO

A primeira tela do software aplicativo desenvolvido que o usuário observará está na Figura 15, a seguir:

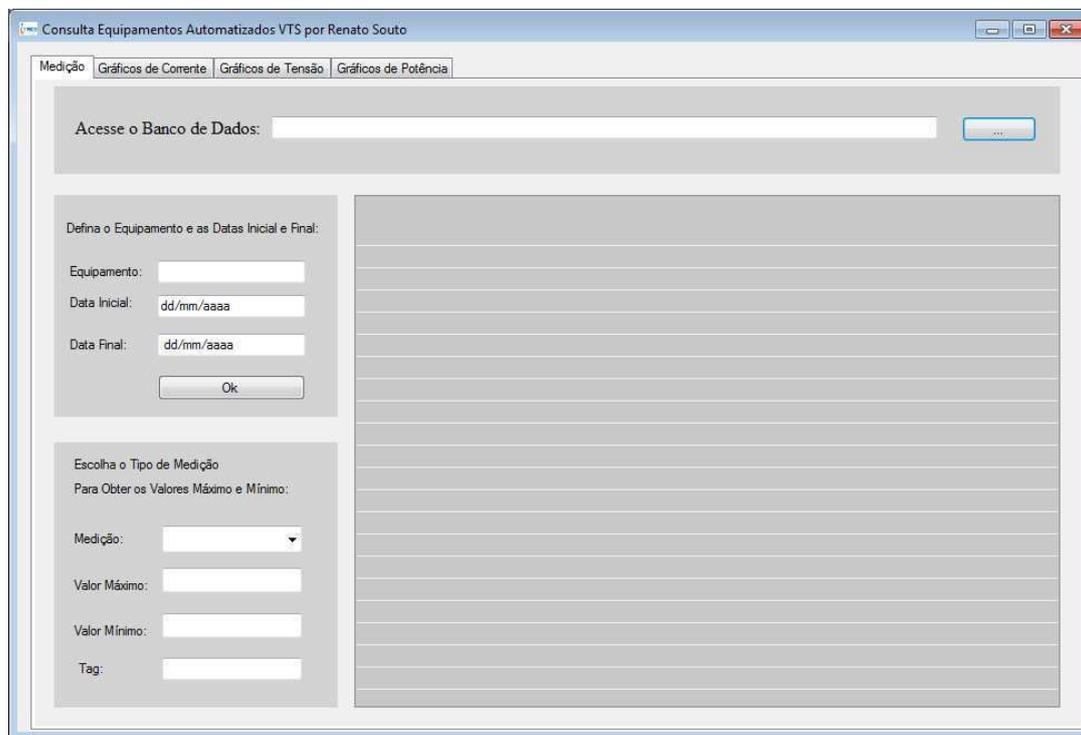


Figura 15: Tela inicial da aplicação

O usuário terá a oportunidade de escolher qual banco de dados o mesmo deseja acessar. Tal opção foi criada para maior flexibilidade do software aplicativo, tendo em vista que usuário pode ter acesso a arquivos de backup. E além de poder ser utilizado pelo COD, o software aplicativo pode ser utilizado pelo COS (Centro de Operação de Subestação).

Depois de ter escolhido banco de dados, o usuário informará qual equipamento deseja consultar e em qual intervalo de tempo. Depois de preenchidos as caixas de texto com as respectivas informações, o usuário clicará no botão “OK”. Então, surgirá uma tabela com as informações das medições nas três fases de tensões e corrente, e das potências ativas e reativas, dentro do intervalo de tempo mencionado. Como um exemplo, tem-se o religador BTF_LPT, cuja identificação é BTF_LPT, com datas de medições, inicial e final, 20/08/2010 e 22/08/2010, respectivamente. Obtém-se, como resultado, uma tabela apresentada na Figura 16.

DATA	HORA	DST_...							
20/08/2...	00:00:00	13,393	13,183	13,368	21	19	21	362	304
20/08/2...	00:15:00	13,454	13,308	13,475	21	19	20	344	305
20/08/2...	00:30:00	13,454	13,308	13,475	21	19	20	344	305
20/08/2...	00:45:00	13,454	13,308	13,475	21	19	20	344	305
20/08/2...	01:00:00	13,454	13,308	13,475	21	19	20	344	305
20/08/2...	01:15:00	13,386	13,309	13,47	20	18	19	331	293
20/08/2...	01:30:00	13,386	13,309	13,47	20	18	19	331	293
20/08/2...	01:45:00	13,386	13,309	13,47	20	18	19	331	293
20/08/2...	02:00:00	13,386	13,309	13,47	20	18	19	331	293
20/08/2...	02:15:00	13,343	13,264	13,417	20	18	19	324	288
20/08/2...	02:30:00	13,343	13,264	13,417	20	18	19	324	288
20/08/2...	02:45:00	13,343	13,264	13,417	20	18	19	324	288
20/08/2...	03:00:00	13,343	13,264	13,417	20	18	19	324	288
20/08/2...	03:15:00	13,156	13,12	13,316	19	18	18	319	268
20/08/2...	03:30:00	13,156	13,12	13,316	19	18	18	319	268
20/08/2...	03:45:00	13,156	13,12	13,316	19	18	18	319	268
20/08/2...	04:00:00	13,156	13,12	13,316	19	18	18	319	268
20/08/2...	04:15:00	13,032	12,992	13,179	20	18	19	328	266
20/08/2...	04:30:00	13,032	12,992	13,179	20	18	19	328	266
20/08/2...	04:45:00	13,032	12,992	13,179	20	18	19	328	266
20/08/2...	05:00:00	13,032	12,992	13,179	20	18	19	328	266
20/08/2...	05:15:00	13,11	13,098	13,278	21	18	19	349	266

Figura 16: Tabela de medições da aplicação

Outra necessidade observada foi com relação aos valores máximos e mínimos de cada medição dentro do intervalo de tempo de interesse. As opções de medição são:

- Tensão AB.
- Tensão BC.
- Tensão CA.
- Corrente IA.
- Corrente IB.
- Corrente IC.
- Potência Ativa
- Potência Reativa.

O usuário escolhe qual informação deseja obter clicando no ícone de opções, então, surgirão os valores nas caixas de textos correspondentes ao valor máximo e mínimo. Na Figura 17, é apresentado um exemplo em que é escolhido o valor máximo e mínimo, da opção “Tensão AB”, e obtido os valores 13,339 KV e 12,621 KV, respectivamente. A Tag do equipamento selecionado também é informada na respectiva caixa de texto.

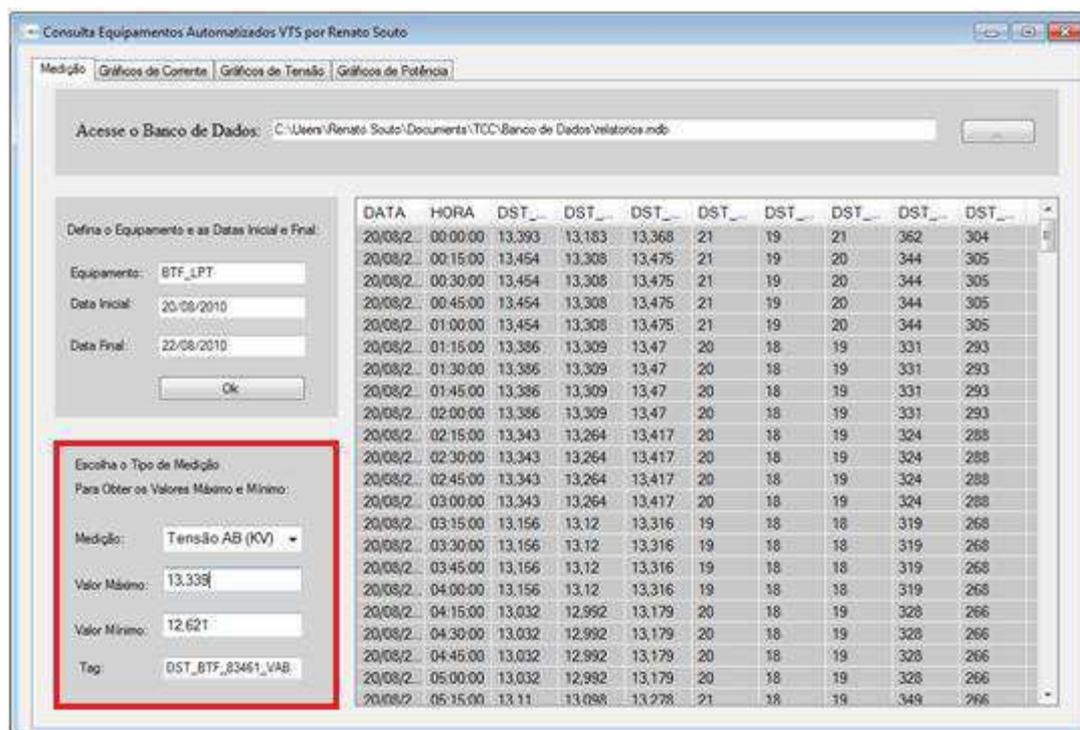


Figura 17: Aplicação com informação do valor máximo e mínimo da medição

Outra opção, para o usuário, são os gráficos das tensões, correntes e potências. Nestes gráficos, é possível visualizar os dados correspondentes a cada medição. Clicando no botão “Gerar Gráfico, obtêm-se os gráficos correspondentes. Na Figura 18, tem-se a representação do gráfico correspondente as medições de corrente, com os dados utilizados no exemplo anterior:

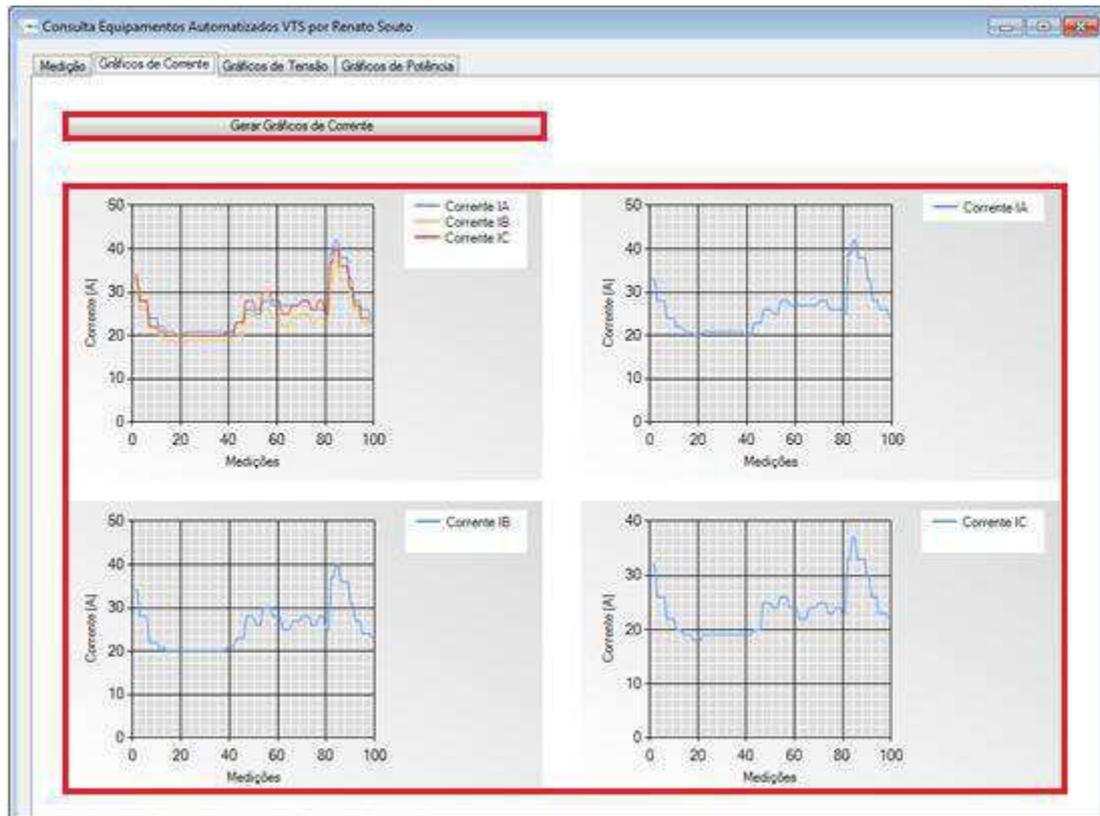


Figura 18: Gráficos de medições de correntes

Nesta Figura, pode-se observar que foram gerados três gráficos. O primeiro gráfico, com as informações das correntes nas três fases, e os demais com as informações das correntes em cada fase. Nas Figuras 19 e 20, têm-se os gráficos de tensão e potência, respectivamente, do exemplo anterior.

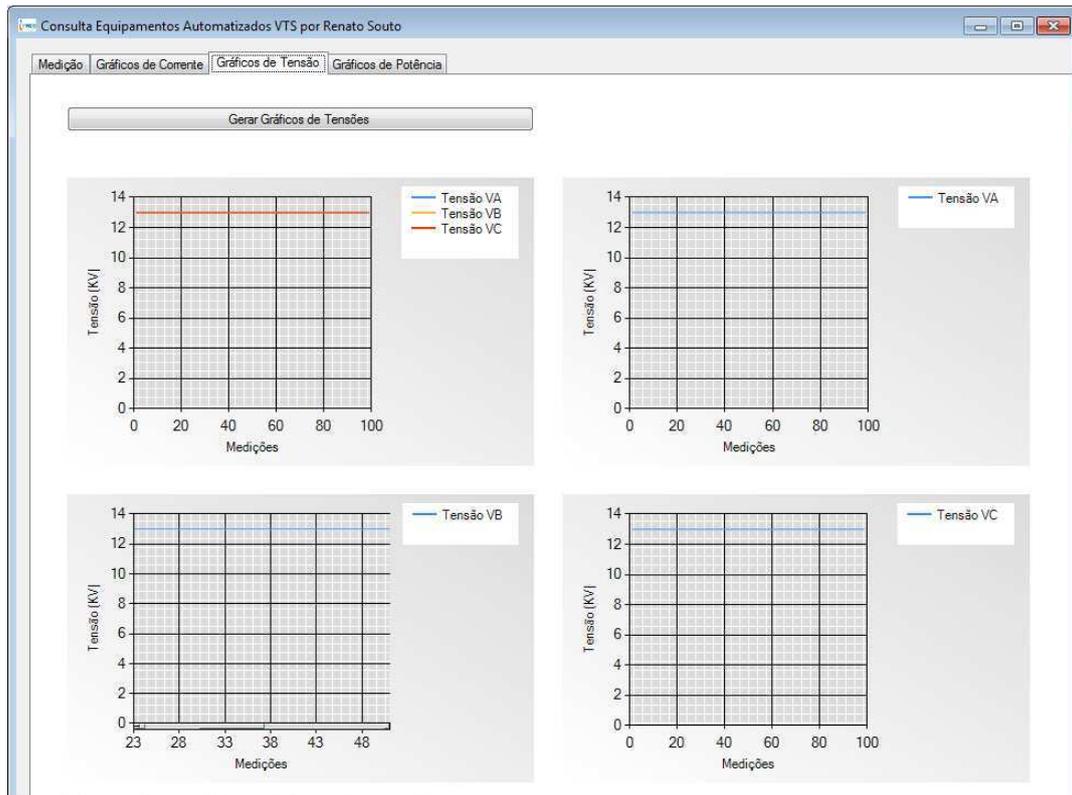


Figura 19: Gráficos de medições de tensões

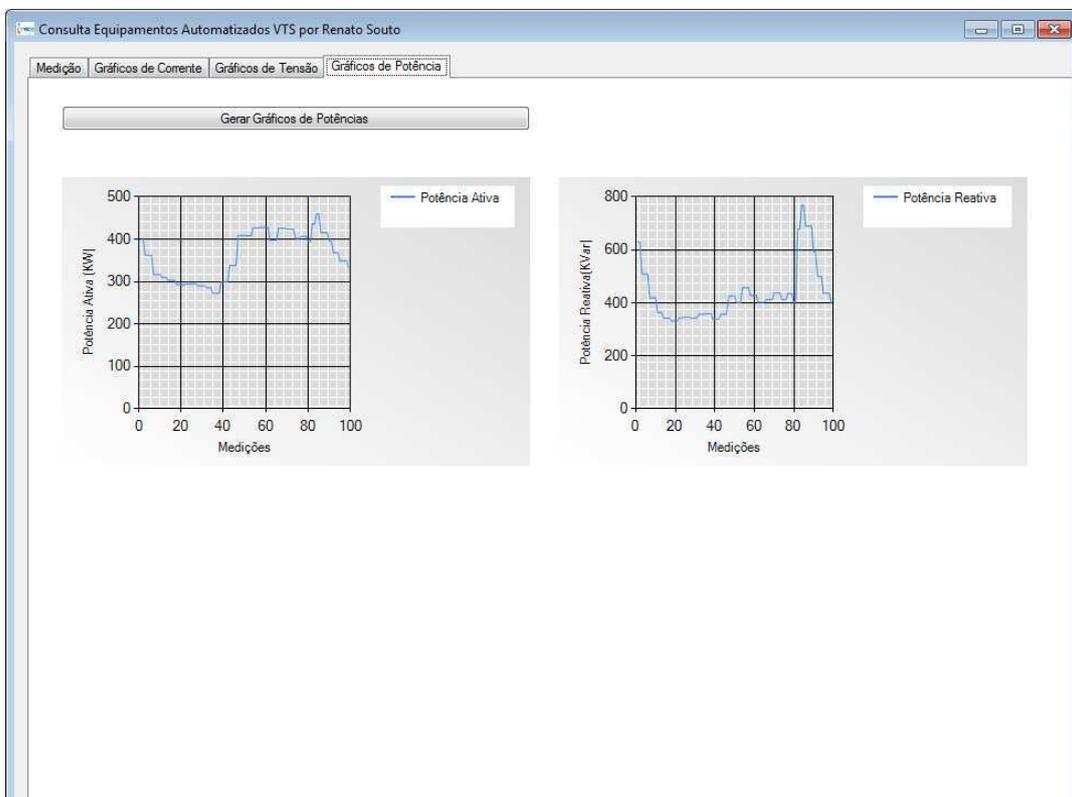


Figura 20: Gráficos de medições de potências

4 ANÁLISE DOS RESULTADOS

Nesta seção será realizada uma análise sobre o funcionamento do software aplicativo e posteriormente uma análise dos resultados obtidos em algumas situações, tipicamente demandas por empresas do setor.

4.1 EXEMPLO 1: TENSÃO EM REGIME PERMANENTE

De acordo com os índices de qualidade PRODIST e de acordo com as Tabela 1, será feita uma análise a respeito das faixas de tensão de atendimento. Na Figura 21, tem-se um exemplo do resultado da busca de informações sobre o equipamento BTF_LPT, numa tabela, entre os dias 20/08/2010 e 22/08/2010, através do software aplicativo.

Consulta Equipamentos Automatizados VTS por Renato Souto

Medição: Gráficos de Corrente | Gráficos de Tensão | Gráficos de Potência

Acesse o Banco de Dados: C:\Users\Renato Souto\Documents\TCC\Banco de Dados\relatorios.mdb

Defina o Equipamento e as Datas Inicial e Final:

Equipamento: BTF_LPT

Data Inicial: 20/08/2010

Data Final: 22/08/2010

Escolha o Tipo de Medição

Para Obter os Valores Máximo e Mínimo:

Medição: Tensão AB (KV)

Valor Máximo: 13,339

Valor Mínimo: 12,621

DATA	HORA	DST_BTF_83461_VAB	DST_1	DST_2	DST_3	DST_4	DST_5
21/08/2010	14:00:00	13,142	13,143	13,243	27	24	27
21/08/2010	14:15:00	13,078	13,121	13,214	28	25	28
21/08/2010	14:30:00	13,078	13,121	13,214	28	25	28
21/08/2010	14:45:00	13,078	13,121	13,214	28	25	28
21/08/2010	15:00:00	13,078	13,121	13,214	28	25	28
21/08/2010	15:15:00	13,228	13,323	13,253	26	23	26
21/08/2010	15:30:00	13,228	13,323	13,253	26	23	26
21/08/2010	15:45:00	13,228	13,323	13,253	26	23	26
21/08/2010	16:00:00	13,114	13,156	13,14	26	24	28
21/08/2010	16:15:00	13,114	13,156	13,14	26	24	28
21/08/2010	16:30:00	13,114	13,156	13,14	26	24	28
21/08/2010	16:45:00	13,206	13,216	13,286	25	23	25
21/08/2010	17:00:00	13,206	13,216	13,286	25	23	25
21/08/2010	17:15:00	13,206	13,216	13,286	25	23	25
21/08/2010	17:30:00	13,206	13,216	13,286	25	23	25
21/08/2010	17:45:00	13,206	13,216	13,286	25	23	25
21/08/2010	18:00:00	12,621	13,082	12,789	39	33	37
21/08/2010	18:15:00	12,621	13,082	12,789	39	33	37
21/08/2010	18:30:00	12,704	13,297	12,971	42	37	40
21/08/2010	18:45:00	12,704	13,297	12,971	42	37	40
21/08/2010	19:00:00	12,769	13,153	12,984	38	33	36

Figura 21: Tela inicial da aplicação

No horário das 18:00hs, o valor da tensão é 12,621 KV. Partindo como tensão de referência o valor de 13,8 KV, e realizando uma análise da tensão em regime

permanente, já que o valor da tensão não se altera nos quinze minutos seguintes, tem-se que:

$$TL = 12,621 \text{ KV}, \quad TR = 13,8 \text{ KV} \quad (10)$$

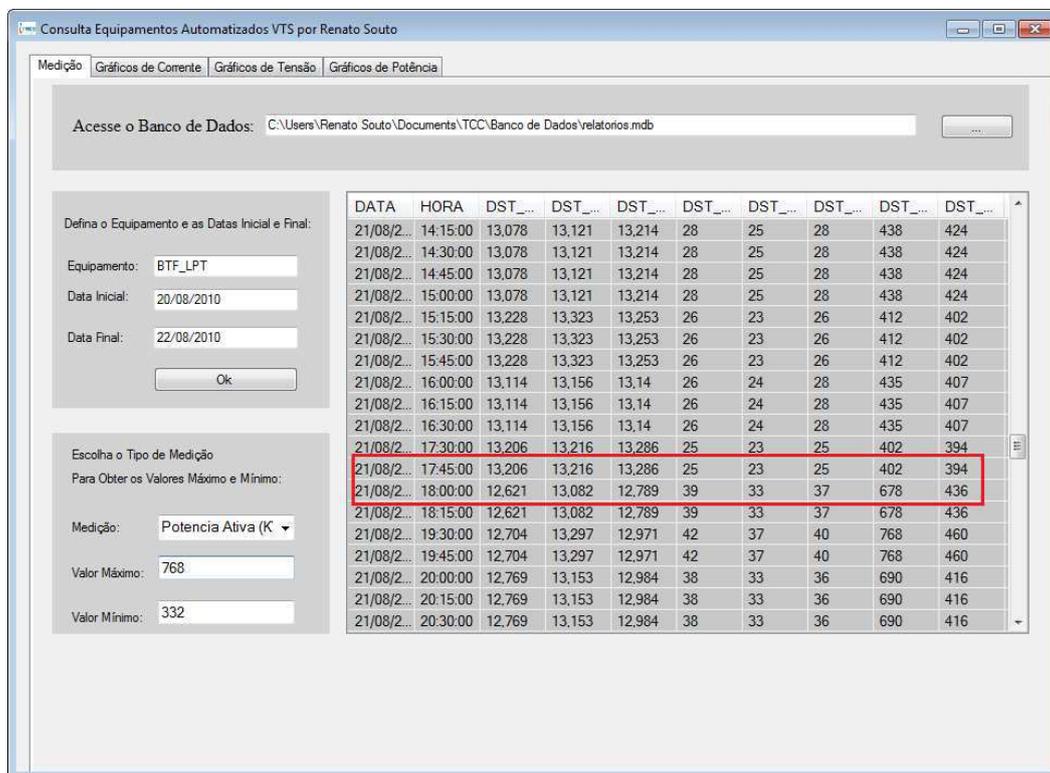
$\frac{TL}{TR} = 0,91$, ou seja, de acordo com a Tabela 1, a tensão de atendimento neste horário é precária.

Já no período da tarde, a situação é diferente. Tomando-se como valor, a tensão referente ao horário das 16:00hs, ou seja, $TL = 13,114 \text{ KV}$. Logo, $\frac{TL}{TR} = 0,95$, Portanto, a tensão de atendimento é adequada, de acordo com as faixas da Tabela 1.

Se tal problema for cotidiano, então, é necessária a instalação de um possível regulador de tensão automático, para assim manter a tensão em horários de pico em níveis aceitáveis.

4.2 EXEMPLO 2: FATOR DE POTÊNCIA

A partir dos dados mostrados na Figura 22, é possível extrair os valores das potências ativa e reativa nas últimas duas colunas da tabela.



Consulta Equipamentos Automatizados VTS por Renato Souto

Medição | Gráficos de Corrente | Gráficos de Tensão | Gráficos de Potência

Acesse o Banco de Dados: C:\Users\Renato Souto\Documents\TCC\Banco de Dados\relatorios.mdb

Defina o Equipamento e as Datas Inicial e Final:

Equipamento: BTF_LPT

Data Inicial: 20/08/2010

Data Final: 22/08/2010

Escolha o Tipo de Medição

Para Obter os Valores Máximo e Mínimo:

Medição: Potencia Ativa (K)

Valor Máximo: 768

Valor Mínimo: 332

DATA	HORA	DST_...							
21/08/2...	14:15:00	13,078	13,121	13,214	28	25	28	438	424
21/08/2...	14:30:00	13,078	13,121	13,214	28	25	28	438	424
21/08/2...	14:45:00	13,078	13,121	13,214	28	25	28	438	424
21/08/2...	15:00:00	13,078	13,121	13,214	28	25	28	438	424
21/08/2...	15:15:00	13,228	13,323	13,253	26	23	26	412	402
21/08/2...	15:30:00	13,228	13,323	13,253	26	23	26	412	402
21/08/2...	15:45:00	13,228	13,323	13,253	26	23	26	412	402
21/08/2...	16:00:00	13,114	13,156	13,14	26	24	28	435	407
21/08/2...	16:15:00	13,114	13,156	13,14	26	24	28	435	407
21/08/2...	16:30:00	13,114	13,156	13,14	26	24	28	435	407
21/08/2...	17:30:00	13,206	13,216	13,286	25	23	25	402	394
21/08/2...	17:45:00	13,206	13,216	13,286	25	23	25	402	394
21/08/2...	18:00:00	12,621	13,082	12,789	39	33	37	678	436
21/08/2...	18:15:00	12,621	13,082	12,789	39	33	37	678	436
21/08/2...	19:30:00	12,704	13,297	12,971	42	37	40	768	460
21/08/2...	19:45:00	12,704	13,297	12,971	42	37	40	768	460
21/08/2...	20:00:00	12,769	13,153	12,984	38	33	36	690	416
21/08/2...	20:15:00	12,769	13,153	12,984	38	33	36	690	416
21/08/2...	20:30:00	12,769	13,153	12,984	38	33	36	690	416

Figura 22: Tabela com os dados das medições

No instante 17:45hs, o valor da potência ativa e reativa é 402 KW e 394 KVar, respectivamente. A partir destes valores, foi possível determinar o fator de potência correspondente:

$$fp = \frac{P}{\sqrt{P^2+Q^2}} = 0,71 , \quad (11)$$

Ou seja, abaixo de 0,92.

Observamos que as 18:00hs, o valor da potência ativa aumenta substancialmente, então, para $P = 678$ KW e $Q = 436$ KVar, temos:

$$fp = \frac{P}{\sqrt{P^2+Q^2}} = 0,87 , \quad (12)$$

Então, a solução proposta é a inserção de um banco de capacitores para que o fator de potência esteja num valor aceitável.

4.3 EXEMPLO 3: ANÁLISE DE INDICADOR DE CONTINUIDADE

Esta análise é inviável pela atual instabilidade de comunicação dos equipamentos da companhia. Na Figura 23, é possível observar que os valores das tensões nos gráficos constam como zero em algumas situações. Provavelmente, este problema ocorreu pela instabilidade de comunicação dos equipamentos, inviabilizando a confiabilidade de uma análise de continuidade de fornecimento de energia. Contudo, pode-se analisar os equipamentos que possuem comunicação instável e realizar um planejamento para manutenção dos mesmos.

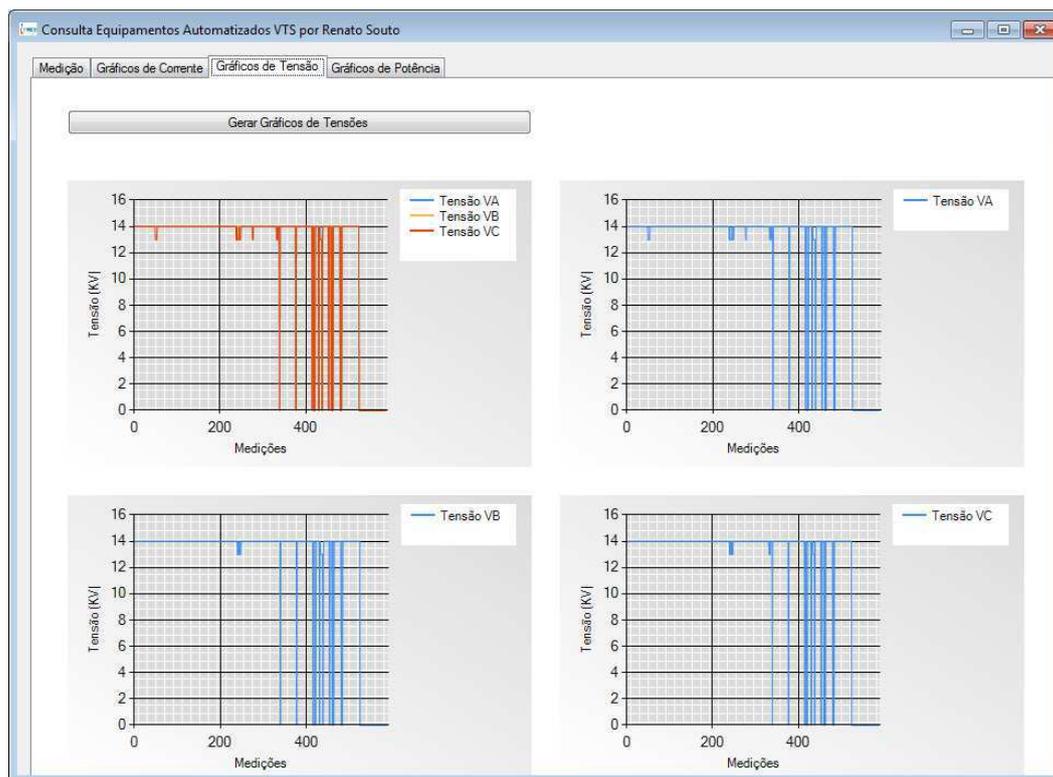


Figura 23: Gráficos dos valores de tensões com comunicação instável

4.4 EXEMPLO 4: FATOR DE CARGA

O fator de carga pode ser calculado observando o valor máximo de potência num determinado período a ser analisado, e calculado o valor médio da demanda neste intervalo.

Na Figura 24, têm-se gráficos de potências, e dependendo da especificação requisitada pelo usuário, o software aplicativo poderá gerar a informação do fator de carga, assim como também do fator de demanda, na própria aplicação. Atualmente, este software não apresenta esta funcionalidade, mas possui os dados que dão subsídios para gerar tal informação.

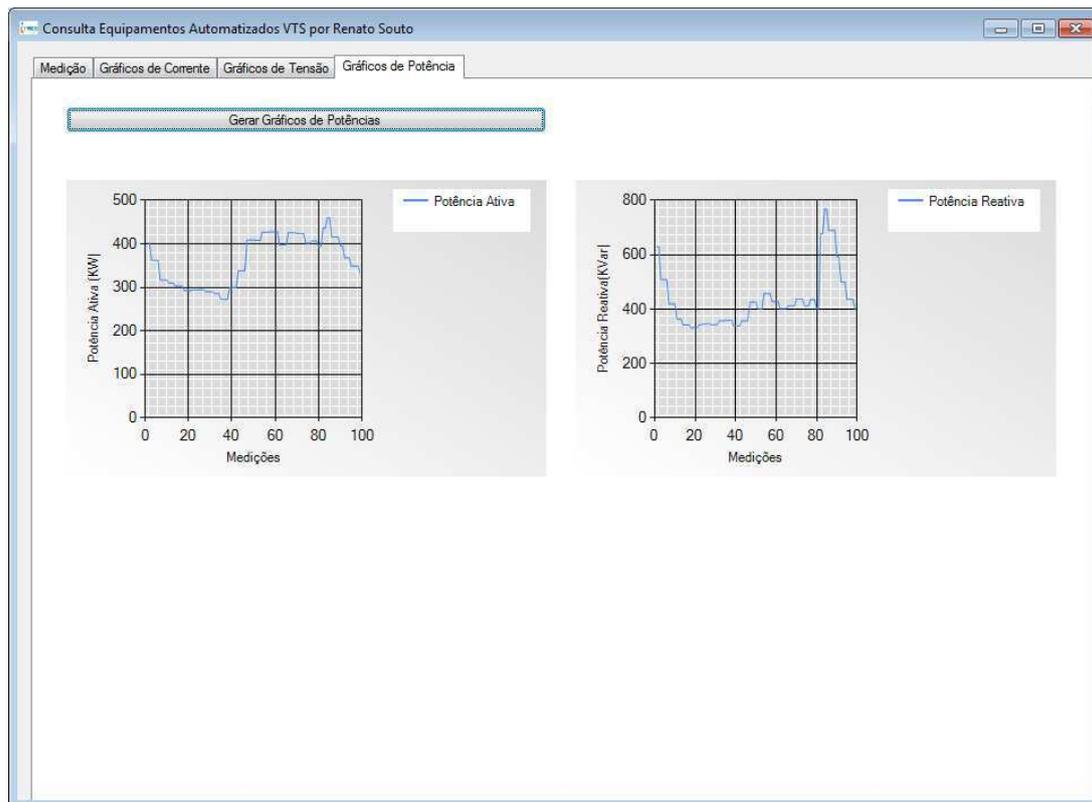


Figura 24: Gráficos de potência ativa e reativa

4.5 EXEMPLO 5: DESEQUILÍBRIO DE FASES

O desequilíbrio de fases gera perdas de energia causadas pelo aquecimento dos equipamentos. Para evitar que esse problema ocorra, é necessário um planejamento e um monitoramento das cargas. Na Figura 25, pode-se observar um desequilíbrio na fase C, onde esta se encontra com a tensão menor em relação às outras fases.

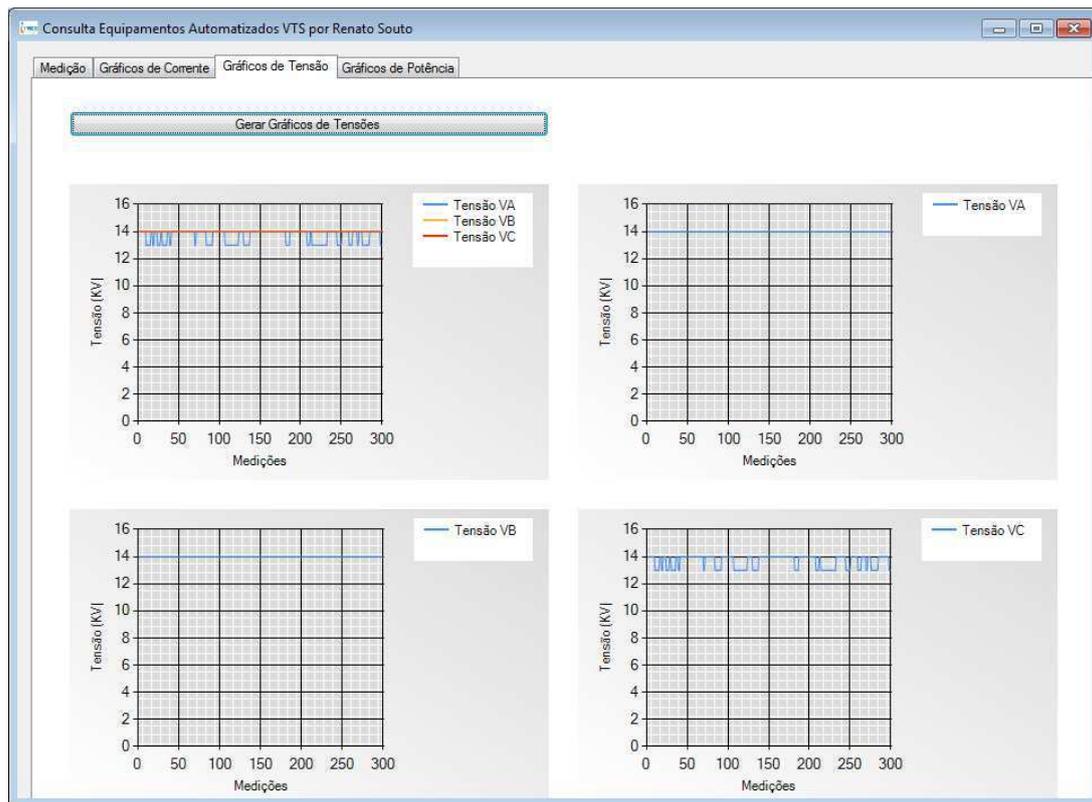


Figura 25: Gráficos de desequilíbrio de fases

Uma maneira de solucionar esse problema seria redistribuir as cargas da forma homogênea, para assim reduzir as perdas no sistema de distribuição ao máximo.

5 CONCLUSÃO

Este trabalho de conclusão de curso tem como objetivo a consolidação dos conhecimentos teóricos adquiridos durante os anos de estudo na universidade. Durante o estágio integrado na Empresa Energisa Paraíba, fui capaz de identificar uma possível melhoria na aquisição de informações com relação aos equipamentos automatizados. Esta percepção de identificar falhas e propor soluções foi adquirida durante os anos de vida acadêmica.

Com relação aos resultados obtidos, estes foram satisfatórios, mesmo diante da limitação de algumas informações sobre o VTS, falta de alguns dados e problemas com relação a instabilidade de comunicação dos equipamentos. Esta aplicação ainda pode ser aperfeiçoada e utilizada no âmbito profissional, afinal, esta versão é apenas um protótipo que pode adquirir inúmeras funcionalidades, além das apresentadas neste trabalho de conclusão de curso.

A princípio, este software foi projetado para que as informações pudessem ser acessadas remotamente, e não apenas na empresa. Mas alguns problemas relacionados a aspectos burocráticos, e o pouco de tempo de estágio, inviabilizam a conclusão desta proposta. Contudo, os resultados obtidos foram significativos.

Os conhecimentos teóricos adquiridos nas disciplinas de Distribuição de Energia Elétrica, Proteção de Sistemas Elétricos, Equipamentos Elétricos e Técnicas de Programação, foram de suma importância para o sucesso no desenvolvimento da aplicação.

BIBLIOGRAFIA

- [1] GIGUER, S. Proteção de Sistemas de Distribuição. Porto Alegre: Sagra, 1988.

- [2] Manual técnico para religadores U27-12 com controladores baseados em CAPM-4 e CAPM-5. Nulec PTY, 2001.

- [3] KIMURA, Bruno. Apostila Rápida Religadores Noja Power. RMS Electric. São Paulo, 2009.

- [4] SHANNON, Natalie. VTS Operators Guide. Trihedral Engineering Limited. Nova Scotia, Canada, 2004.

- [5] <http://www.macoratti.net/Default.aspx>. Acesso em Novembro de 2010.

- [6] <http://msdn.microsoft.com/default.aspx>. Acesso em Novembro de 2010.

- [7] ANEEL. Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST) - Módulo 8 – Qualidade da Energia Elétrica. 2010.

- [8] SOUZA, B. Distribuição de Energia Elétrica. Campina Grande, 1988.

APÊNDICE A - CÓDIGO DA APLICAÇÃO

```
//Forms.cs

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;
using System.Windows.Forms.DataVisualization.Charting;

namespace ConsultaEquipamentosVTS
{
    public partial class Form1 : Form
    {
        private OleDbConnection conexaoBD;
        private string arquivoDados;
        private string campo1;

        string horaInicial = "21:00:00";
        string horaFinal = "22:45:00";

        public Form1()
        {
            InitializeComponent();
        }

        //Preencher gráficos de Corrente
        private void tabPage2_Click(object sender, EventArgs e)
        {

        }

        private void tabPage1_Click(object sender, EventArgs e)
        {

        }

        private void conexaoDados(string arquivoDadosBD)
        {

            //define a string de conexão usando o provedor OleDb
            string conexao =
@"provider=microsoft.jet.oledb.4.0;Password=" + arquivoDadosBD;

            try
            {
                //Inicializa o objeto conexaoBD
                conexaoBD = new OleDbConnection(conexao);
            }
            catch (Exception ex)

```

```

        {
            MessageBox.Show(ex.Message);
        }

        //preencherTreeView();
    }

private void textBox1_TextChanged(object sender, EventArgs e)
{
}

private void label1_Click(object sender, EventArgs e)
{
}

private void txtBancoDados_TextChanged(object sender, EventArgs e)
{
}

private string selecionaArquivoBD(string pastaInicial)
{
    //abre uma nova caixa de dialogo para selecionar arquivo
    OpenFileDialog dialog = new OpenFileDialog();
    //define alguns parâmetros
    dialog.Filter = "mdb files (*.mdb)|*.mdb|All files (*.*)|*.*";
    dialog.InitialDirectory = pastaInicial;
    dialog.Title = "Selecione um Arquivo";
    //retorna o arquivo selecionado
    return (dialog.ShowDialog() == DialogResult.OK) ? dialog.FileName :
null;
}

private void Form1_Load(object sender, EventArgs e)
{
    lvDados.Clear();
    lvDados.View = View.Details;
    lvDados.LabelEdit = false;
    lvDados.FullRowSelect = true;
    lvDados.GridLines = true;
}

private void lvDados_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void PreencherListView(OleDbConnection conBD, string nomeTabela,
string minhaDataFormatada1, string minhaDataFormatada2)
{
    OleDbCommand cmdBD; //cmdBD2;
    OleDbDataReader datReader;

    string strCampo = "SELECT * FROM [" + nomeTabela + "] WHERE DATA
BETWEEN #" + minhaDataFormatada1 + "# AND #" + minhaDataFormatada2 + "#";

```

```

    // Inicializa o objeto command
    cmdBD = new OleDbCommand(strCampo, conBD);
    //abre a conexão e executa o command
    conBD.Open();
    datReader = cmdBD.ExecuteReader();

    for (int c = 0; c < datReader.FieldCount; c++)
    {
        ColumnHeader itemColuna = new ColumnHeader();
        itemColuna.Text = datReader.GetName(c);
        lvDados.Columns.Add(itemColuna);
    }

    //define um item listview
    ListViewItem item;

    //inicia leitura do datareader
    while (datReader.Read())
    {
        item = new ListViewItem();
        item.Text = datReader.GetValue(0).ToString();

        //preenche o listview com itens
        for (int c = 1; c < datReader.FieldCount; c++)
        {
            item.SubItems.Add(datReader.GetValue(c).ToString());
        }

        lvDados.Items.Add(item);
    }
    datReader.Close();
    conBD.Close();
}

private void buttonOk_Click(object sender, EventArgs e)
{
    string bEquipamento = buscaEquipamento.Text;
    lvDados.Clear();

    PreencherListView(conexaoBD, buscaEquipamento.Text, textBox2.Text,
textBox3.Text); //bEquipamento); // "BTF_LPT");
    valorMaximo.Text = "";
    valorMinimo.Text = "";
}

private void textBox3_TextChanged(object sender, EventArgs e)
{
    // ColumnHeader ch;
}

private string dadosResultadoMaximo(OleDbConnection conBDCampo, string
nomeTabela, string campo, string minhaDataFormatada1, string minhaDataFormatada2)
{
    OleDbCommand cmdBDCampo;
    OleDbDataReader datReaderCampo;
    string valorResultado = "";
    string campoTabela;

```

```

        textBox4.Text = campo;

        campoTabela = "SELECT max([" + campo + "]) FROM [" + nomeTabela + "]
WHERE DATA & HORA BETWEEN #" + minhaDataFormatada1 + "# AND #" +
minhaDataFormatada2 + "# & #" + horaInicial + "# AND #" + horaFinal + "#";

        cmdBDCampo = new OleDbCommand(campoTabela, conBDCampo);
        //abre a conexão e executa o command
        conBDCampo.Open();
        datReaderCampo = cmdBDCampo.ExecuteReader();

        // Preenche o ListView
        while (datReaderCampo.Read())
        {
            valorResultado = datReaderCampo.GetValue(0).ToString();

        }
        //fecha o datareader e a conexao
        datReaderCampo.Close();
        conBDCampo.Close();

        return valorResultado;
    }

    private string dadosResultadoMinimo(OleDbConnection conBDCampo, string
nomeTabela, string campo, string minhaDataFormatada1, string minhaDataFormatada2)
    {
        OleDbCommand cmdBDCampo;
        OleDbDataReader datReaderCampo;
        string valorResultado = "";
        string campoTabela;

        textBox4.Text = campo;

        campoTabela = "SELECT min([" + campo + "]) FROM [" + nomeTabela + "]
WHERE DATA & HORA BETWEEN #" + minhaDataFormatada1 + "# AND #" +
minhaDataFormatada2 + "# & #" + horaInicial + "# AND #" + horaFinal + "#";

        cmdBDCampo = new OleDbCommand(campoTabela, conBDCampo);
        //abre a conexão e executa o command
        conBDCampo.Open();
        datReaderCampo = cmdBDCampo.ExecuteReader();

        // Preenche o ListView

        while (datReaderCampo.Read())
        {
            valorResultado = datReaderCampo.GetValue(0).ToString();

        }
        //fecha o datareader e a conexao
        datReaderCampo.Close();
        conBDCampo.Close();

        return valorResultado;
    }

    private string buscaCampo(OleDbConnection conBD, string nomeTabela, int
coluna)

```

```

{
    OleDbCommand cmdBD; //cmdBD2;
    OleDbDataReader datReader;

    string strCampo = "SELECT * FROM [" + nomeTabela + "]";

    // Inicializa o objeto command
    cmdBD = new OleDbCommand(strCampo, conBD);
    //abre a conexão e executa o command
    conBD.Open();
    datReader = cmdBD.ExecuteReader();

    textBox4.Text = datReader.GetName(coluna);

    datReader.Close();
    conBD.Close();

    return textBox4.Text;
}

private void btnSelecionaBD_Click(object sender, EventArgs e)
{
    arquivoDados = selecionaArquivoBD("c:\\dados\\");
    conexaoDados(conexaoDados);
    txtBancoDados.Text = arquivoDados;
}

private void comboBox1_SelectedIndexChanged_1(object sender, EventArgs e)
{
    ComboBox comboBox = (ComboBox)sender;

    string item = (string)comboBox.SelectedItem;
    int c;

    switch (item)
    {
        case "Tensão AB (KV)":

            c = 2;
            campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, c);
            valorMaximo.Text = dadosResultadoMaximo(conexaoBD,
            buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
            valorMinimo.Text = dadosResultadoMinimo(conexaoBD,
            buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
            break;
        case "Tensão BC (KV)":

            c = 3;
            campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, c);
            valorMaximo.Text = dadosResultadoMaximo(conexaoBD,
            buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
            valorMinimo.Text = dadosResultadoMinimo(conexaoBD,
            buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
            break;
        case "Tensão CA (KV)":
    }
}

```

```

        c = 4;
        campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, c);
        valorMaximo.Text = dadosResultadoMaximo(conexaoBD,
        buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
        valorMinimo.Text = dadosResultadoMinimo(conexaoBD,
        buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
        break;
        case "Corrente IA (A)":

            c = 5;
            campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, c);
            valorMaximo.Text = dadosResultadoMaximo(conexaoBD,
            buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
            valorMinimo.Text = dadosResultadoMinimo(conexaoBD,
            buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
            break;
            case "Corrente IB (A)":

                c = 6;
                campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, c);
                valorMaximo.Text = dadosResultadoMaximo(conexaoBD,
                buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
                valorMinimo.Text = dadosResultadoMinimo(conexaoBD,
                buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
                break;
                case "Corrente IC (A)":

                    c = 7;
                    campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, c);
                    valorMaximo.Text = dadosResultadoMaximo(conexaoBD,
                    buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
                    valorMinimo.Text = dadosResultadoMinimo(conexaoBD,
                    buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
                    break;
                    case "Potencia Ativa (KW)":

                        c = 8;
                        campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, c);
                        valorMaximo.Text = dadosResultadoMaximo(conexaoBD,
                        buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
                        valorMinimo.Text = dadosResultadoMinimo(conexaoBD,
                        buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
                        break;
                        case "Potencia Reativa (KVar)":

                            c = 9;
                            campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, c);
                            valorMaximo.Text = dadosResultadoMaximo(conexaoBD,
                            buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
                            valorMinimo.Text = dadosResultadoMinimo(conexaoBD,
                            buscaEquipamento.Text, campo1, textBox2.Text, textBox3.Text);
                            break;
                    }
            }

        }

private void label7_Click(object sender, EventArgs e)
{

```

```

    }

    private void chart1_Click(object sender, EventArgs e)
    {
    }

    private void PreencherGrafico(int tipoMedicao, Chart chart,
OleDbConnection conBDCampo, string nomeTabela, string campo, string
minhaDataFormatada1, string minhaDataFormatada2, int serie)
    {
        OleDbCommand cmdBDCampo;
        OleDbDataReader datReaderCampo;
        string campoTabela;

        campoTabela = "SELECT (" + campo + ") FROM [" + nomeTabela + "]
WHERE DATA & HORA BETWEEN #" + minhaDataFormatada1 + "# AND #" +
minhaDataFormatada2 + "# & #" + horaInicial + "# AND #" + horaFinal + "#";

        cmdBDCampo = new OleDbCommand(campoTabela, conBDCampo);

        //abre a conexão e executa o command
        conBDCampo.Open();
        datReaderCampo = cmdBDCampo.ExecuteReader();

        //tipo de medicao: corrente, tensao ou potencia
        if (tipoMedicao == 1)
        {
            chart.Series[0].Points.Clear();
            chart1.Series[serie].Points.Clear();

            while (datReaderCampo.Read())
            {

chart.Series[0].Points.Add(Convert.ToInt32(datReaderCampo.GetValue(0)));

chart1.Series[serie].Points.Add(Convert.ToInt32(datReaderCampo.GetValue(0)));
            }
        }

        if (tipoMedicao == 2)
        {
            chart.Series[0].Points.Clear();
            chart5.Series[serie].Points.Clear();

            while (datReaderCampo.Read())
            {

chart.Series[0].Points.Add(Convert.ToInt32(datReaderCampo.GetValue(0)));

chart5.Series[serie].Points.Add(Convert.ToInt32(datReaderCampo.GetValue(0)));
            }
        }

        if (tipoMedicao == 3)
        {
            chart.Series[0].Points.Clear();

            while (datReaderCampo.Read())
            {

```

```

chart.Series[0].Points.Add(Convert.ToInt32(datReaderCampo.GetValue(0)));
    }
}

//fecha o datareader e a conexao
datReaderCampo.Close();
conBDCampo.Close();

}

private void gerarGraficoCorrente_Click(object sender, EventArgs e)
{
    campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, 5);
    PreencherGrafico(1, chart2, conexaoBD, buscaEquipamento.Text, campo1,
textBox2.Text, textBox3.Text, 0);

    campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, 6);
    PreencherGrafico(1, chart3, conexaoBD, buscaEquipamento.Text, campo1,
textBox2.Text, textBox3.Text, 1);

    campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, 7);
    PreencherGrafico(1, chart4, conexaoBD, buscaEquipamento.Text, campo1,
textBox2.Text, textBox3.Text, 2);
}

private void gerarGraficoTensao_Click(object sender, EventArgs e)
{
    campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, 2);
    PreencherGrafico(2, chart6, conexaoBD, buscaEquipamento.Text, campo1,
textBox2.Text, textBox3.Text, 0);

    campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, 3);
    PreencherGrafico(2, chart7, conexaoBD, buscaEquipamento.Text, campo1,
textBox2.Text, textBox3.Text, 1);

    campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, 4);
    PreencherGrafico(2, chart8, conexaoBD, buscaEquipamento.Text, campo1,
textBox2.Text, textBox3.Text, 2);
}

private void gerarGraficoPotencia_Click(object sender, EventArgs e)
{
    campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, 8);
    PreencherGrafico(3, chart9, conexaoBD, buscaEquipamento.Text, campo1,
textBox2.Text, textBox3.Text, 0);

    campo1 = buscaCampo(conexaoBD, buscaEquipamento.Text, 9);
    PreencherGrafico(3, chart10, conexaoBD, buscaEquipamento.Text,
campo1, textBox2.Text, textBox3.Text, 1);
}

private void label5_Click(object sender, EventArgs e)
{

```

```
    }  
    private void textBox1_TextChanged_1(object sender, EventArgs e)  
    {  
    }  
} }  
}
```

```

//Forms1.Designer.cs

namespace ConsultaEquipamentosVTS
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea13
= new System.Windows.Forms.DataVisualization.Charting.ChartArea();
            System.Windows.Forms.DataVisualization.Charting.Legend legend13 = new
System.Windows.Forms.DataVisualization.Charting.Legend();
            System.Windows.Forms.DataVisualization.Charting.LegendItem
legendItem12 = new System.Windows.Forms.DataVisualization.Charting.LegendItem();
            System.Windows.Forms.DataVisualization.Charting.LegendCell
legendCell12 = new System.Windows.Forms.DataVisualization.Charting.LegendCell();
            System.Windows.Forms.DataVisualization.Charting.Series series17 = new
System.Windows.Forms.DataVisualization.Charting.Series();
            System.Windows.Forms.DataVisualization.Charting.Series series18 = new
System.Windows.Forms.DataVisualization.Charting.Series();
            System.Windows.Forms.DataVisualization.Charting.Series series19 = new
System.Windows.Forms.DataVisualization.Charting.Series();
            System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea14
= new System.Windows.Forms.DataVisualization.Charting.ChartArea();
            System.Windows.Forms.DataVisualization.Charting.Legend legend14 = new
System.Windows.Forms.DataVisualization.Charting.Legend();
            System.Windows.Forms.DataVisualization.Charting.LegendItem
legendItem13 = new System.Windows.Forms.DataVisualization.Charting.LegendItem();
            System.Windows.Forms.DataVisualization.Charting.LegendCell
legendCell13 = new System.Windows.Forms.DataVisualization.Charting.LegendCell();
            System.Windows.Forms.DataVisualization.Charting.Series series20 = new
System.Windows.Forms.DataVisualization.Charting.Series();
            System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea15
= new System.Windows.Forms.DataVisualization.Charting.ChartArea();
            System.Windows.Forms.DataVisualization.Charting.Legend legend15 = new
System.Windows.Forms.DataVisualization.Charting.Legend();
            System.Windows.Forms.DataVisualization.Charting.LegendItem
legendItem14 = new System.Windows.Forms.DataVisualization.Charting.LegendItem();

```



```

        System.Windows.Forms.DataVisualization.Charting.Series series15 = new
System.Windows.Forms.DataVisualization.Charting.Series();
        System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea12
= new System.Windows.Forms.DataVisualization.Charting.ChartArea();
        System.Windows.Forms.DataVisualization.Charting.Legend legend12 = new
System.Windows.Forms.DataVisualization.Charting.Legend();
        System.Windows.Forms.DataVisualization.Charting.LegendItem
legendItem11 = new System.Windows.Forms.DataVisualization.Charting.LegendItem();
        System.Windows.Forms.DataVisualization.Charting.LegendCell
legendCell11 = new System.Windows.Forms.DataVisualization.Charting.LegendCell();
        System.Windows.Forms.DataVisualization.Charting.Series series16 = new
System.Windows.Forms.DataVisualization.Charting.Series();
        System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
        this.tabControl1 = new System.Windows.Forms.TabControl();
        this.tabPage1 = new System.Windows.Forms.TabPage();
        this.panel3 = new System.Windows.Forms.Panel();
        this.label1 = new System.Windows.Forms.Label();
        this.txtBancoDados = new System.Windows.Forms.TextBox();
        this.btnSeleccionaBD = new System.Windows.Forms.Button();
        this.panel2 = new System.Windows.Forms.Panel();
        this.label11 = new System.Windows.Forms.Label();
        this.label10 = new System.Windows.Forms.Label();
        this.label7 = new System.Windows.Forms.Label();
        this.label8 = new System.Windows.Forms.Label();
        this.label3 = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.textBox4 = new System.Windows.Forms.TextBox();
        this.valorMaximo = new System.Windows.Forms.TextBox();
        this.comboBox1 = new System.Windows.Forms.ComboBox();
        this.valorMinimo = new System.Windows.Forms.TextBox();
        this.panel1 = new System.Windows.Forms.Panel();
        this.label9 = new System.Windows.Forms.Label();
        this.buttonOk = new System.Windows.Forms.Button();
        this.buscaEquipamento = new System.Windows.Forms.TextBox();
        this.label6 = new System.Windows.Forms.Label();
        this.textBox3 = new System.Windows.Forms.TextBox();
        this.label5 = new System.Windows.Forms.Label();
        this.textBox2 = new System.Windows.Forms.TextBox();
        this.label4 = new System.Windows.Forms.Label();
        this.lvDados = new System.Windows.Forms.ListView();
        this.tabPage2 = new System.Windows.Forms.TabPage();
        this.gerarGraficoCorrente = new System.Windows.Forms.Button();
        this.labelTeste = new System.Windows.Forms.Label();
        this.chart1 = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.chart3 = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.chart4 = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.chart2 = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.tabPage3 = new System.Windows.Forms.TabPage();
        this.chart6 = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.chart7 = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.chart8 = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.gerarGraficoTensao = new System.Windows.Forms.Button();
        this.chart5 = new
System.Windows.Forms.DataVisualization.Charting.Chart();

```

```

        this.tabPage4 = new System.Windows.Forms.TabPage();
        this.chart9 = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.gerarGraficoPotencia = new System.Windows.Forms.Button();
        this.chart10 = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.pictureBox1 = new System.Windows.Forms.PictureBox();
        this.tabControl1.SuspendLayout();
        this.tabPage1.SuspendLayout();
        this.panel3.SuspendLayout();
        this.panel2.SuspendLayout();
        this.panel1.SuspendLayout();
        this.tabPage2.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.chart1)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.chart3)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.chart4)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.chart2)).BeginInit();
        this.tabPage3.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.chart6)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.chart7)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.chart8)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.chart5)).BeginInit();
        this.tabPage4.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.chart9)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.chart10)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
        this.SuspendLayout();
        //
        // tabControl1
        //
        this.tabControl1.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
        | System.Windows.Forms.AnchorStyles.Left)
        | System.Windows.Forms.AnchorStyles.Right)));
        this.tabControl1.Controls.Add(this.tabPage1);
        this.tabControl1.Controls.Add(this.tabPage2);
        this.tabControl1.Controls.Add(this.tabPage3);
        this.tabControl1.Controls.Add(this.tabPage4);
        this.tabControl1.Location = new System.Drawing.Point(12, 12);
        this.tabControl1.Name = "tabControl1";
        this.tabControl1.SelectedIndex = 0;
        this.tabControl1.Size = new System.Drawing.Size(949, 793);
        this.tabControl1.TabIndex = 0;
        //
        // tabPage1
        //
        this.tabPage1.BackColor = System.Drawing.Color.Transparent;
        this.tabPage1.Controls.Add(this.panel3);
        this.tabPage1.Controls.Add(this.panel2);
        this.tabPage1.Controls.Add(this.panel1);

```

```

this.tabPage1.Controls.Add(this.lvDados);
this.tabPage1.Location = new System.Drawing.Point(4, 22);
this.tabPage1.Name = "tabPage1";
this.tabPage1.Padding = new System.Windows.Forms.Padding(3);
this.tabPage1.Size = new System.Drawing.Size(941, 767);
this.tabPage1.TabIndex = 0;
this.tabPage1.Text = "Medição";
this.tabPage1.Click += new System.EventHandler(this.tabPage1_Click);
//
// panel3
//
this.panel3.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right))));
this.panel3.BackColor = System.Drawing.Color.LightGray;
this.panel3.Controls.Add(this.label1);
this.panel3.Controls.Add(this.txtBancoDados);
this.panel3.Controls.Add(this.btnSelecionaBD);
this.panel3.Location = new System.Drawing.Point(17, 6);
this.panel3.Name = "panel3";
this.panel3.Size = new System.Drawing.Size(901, 79);
this.panel3.TabIndex = 39;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Times New Roman", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)0));
this.label1.Location = new System.Drawing.Point(15, 28);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(174, 19);
this.label1.TabIndex = 37;
this.label1.Text = "Acesse o Banco de Dados:";
//
// txtBancoDados
//
this.txtBancoDados.Location = new System.Drawing.Point(195, 27);
this.txtBancoDados.Name = "txtBancoDados";
this.txtBancoDados.Size = new System.Drawing.Size(596, 20);
this.txtBancoDados.TabIndex = 19;
//
// btnSelecionaBD
//
this.btnSelecionaBD.Location = new System.Drawing.Point(813, 27);
this.btnSelecionaBD.Name = "btnSelecionaBD";
this.btnSelecionaBD.Size = new System.Drawing.Size(67, 23);
this.btnSelecionaBD.TabIndex = 20;
this.btnSelecionaBD.Text = "...";
this.btnSelecionaBD.UseVisualStyleBackColor = true;
this.btnSelecionaBD.Click += new
System.EventHandler(this.btnSelecionaBD_Click);
//
// panel2
//
this.panel2.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left))));
this.panel2.BackColor = System.Drawing.Color.LightGray;
this.panel2.Controls.Add(this.label11);

```

```

this.panel2.Controls.Add(this.label10);
this.panel2.Controls.Add(this.label7);
this.panel2.Controls.Add(this.label8);
this.panel2.Controls.Add(this.label3);
this.panel2.Controls.Add(this.label2);
this.panel2.Controls.Add(this.textBox4);
this.panel2.Controls.Add(this.valorMaximo);
this.panel2.Controls.Add(this.comboBox1);
this.panel2.Controls.Add(this.valorMinimo);
this.panel2.Location = new System.Drawing.Point(17, 327);
this.panel2.Name = "panel2";
this.panel2.Size = new System.Drawing.Size(254, 239);
this.panel2.TabIndex = 38;
//
// label11
//
this.label11.AutoSize = true;
this.label11.Location = new System.Drawing.Point(19, 197);
this.label11.Name = "label11";
this.label11.Size = new System.Drawing.Size(29, 13);
this.label11.TabIndex = 41;
this.label11.Text = "Tag:";
//
// label10
//
this.label10.AutoSize = true;
this.label10.Location = new System.Drawing.Point(15, 35);
this.label10.Name = "label10";
this.label10.Size = new System.Drawing.Size(199, 13);
this.label10.TabIndex = 40;
this.label10.Text = "Para Obter os Valores Máximo e Mínimo:";
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(15, 13);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(137, 13);
this.label7.TabIndex = 39;
this.label7.Text = "Escolha o Tipo de Medição";
//
// label8
//
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(15, 80);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(51, 13);
this.label8.TabIndex = 33;
this.label8.Text = "Medição:";
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(15, 163);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(72, 13);
this.label3.TabIndex = 32;
this.label3.Text = "Valor Mínimo:";
//
// label2
//
this.label2.AutoSize = true;

```

```

this.label2.Location = new System.Drawing.Point(15, 122);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(73, 13);
this.label2.TabIndex = 30;
this.label2.Text = "Valor Máximo:";
//
// textBox4
//
this.textBox4.Location = new System.Drawing.Point(97, 194);
this.textBox4.Name = "textBox4";
this.textBox4.Size = new System.Drawing.Size(125, 20);
this.textBox4.TabIndex = 36;
//
// valorMaximo
//
this.valorMaximo.Font = new System.Drawing.Font("Microsoft Sans
Serif", 9.75F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)0));
this.valorMaximo.Location = new System.Drawing.Point(97, 113);
this.valorMaximo.Name = "valorMaximo";
this.valorMaximo.Size = new System.Drawing.Size(125, 22);
this.valorMaximo.TabIndex = 27;
//
// comboBox1
//
this.comboBox1.Font = new System.Drawing.Font("Microsoft Sans Serif",
9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)0));
this.comboBox1.FormattingEnabled = true;
this.comboBox1.Items.AddRange(new object[] {
    "Tensão AB (KV)",
    "Tensão BC (KV)",
    "Tensão CA (KV)",
    "Corrente IA (A)",
    "Corrente IB (A)",
    "Corrente IC (A)",
    "Potencia Reativa (KW)",
    "Potencia Ativa (KVar)"});
this.comboBox1.Location = new System.Drawing.Point(97, 75);
this.comboBox1.Name = "comboBox1";
this.comboBox1.Size = new System.Drawing.Size(125, 24);
this.comboBox1.TabIndex = 28;
this.comboBox1.SelectedIndexChanged += new
System.EventHandler(this.comboBox1_SelectedIndexChanged_1);
//
// valorMinimo
//
this.valorMinimo.Font = new System.Drawing.Font("Microsoft Sans
Serif", 9.75F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)0));
this.valorMinimo.Location = new System.Drawing.Point(97, 154);
this.valorMinimo.Name = "valorMinimo";
this.valorMinimo.Size = new System.Drawing.Size(125, 22);
this.valorMinimo.TabIndex = 29;
//
// panel1
//
this.panel1.BackColor = System.Drawing.Color.LightGray;
this.panel1.Controls.Add(this.label9);
this.panel1.Controls.Add(this.buttonOk);
this.panel1.Controls.Add(this.buscaEquipamento);
this.panel1.Controls.Add(this.label6);

```

```

this.panel1.Controls.Add(this.textBox3);
this.panel1.Controls.Add(this.label5);
this.panel1.Controls.Add(this.textBox2);
this.panel1.Controls.Add(this.label4);
this.panel1.Location = new System.Drawing.Point(17, 104);
this.panel1.Name = "panel1";
this.panel1.Size = new System.Drawing.Size(254, 200);
this.panel1.TabIndex = 37;
//
// label9
//
this.label9.AutoSize = true;
this.label9.Location = new System.Drawing.Point(8, 23);
this.label9.Name = "label9";
this.label9.Size = new System.Drawing.Size(233, 13);
this.label9.TabIndex = 36;
this.label9.Text = "Defina o Equipamento e as Datas Inicial e
Final:";
//
// buttonOk
//
this.buttonOk.Location = new System.Drawing.Point(93, 162);
this.buttonOk.Name = "buttonOk";
this.buttonOk.Size = new System.Drawing.Size(132, 23);
this.buttonOk.TabIndex = 26;
this.buttonOk.Text = "Ok";
this.buttonOk.UseVisualStyleBackColor = true;
this.buttonOk.Click += new System.EventHandler(this.buttonOk_Click);
//
// buscaEquipamento
//
this.buscaEquipamento.Location = new System.Drawing.Point(93, 59);
this.buscaEquipamento.Name = "buscaEquipamento";
this.buscaEquipamento.Size = new System.Drawing.Size(132, 20);
this.buscaEquipamento.TabIndex = 22;
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(11, 128);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(58, 13);
this.label6.TabIndex = 35;
this.label6.Text = "Data Final:";
//
// textBox3
//
this.textBox3.Location = new System.Drawing.Point(93, 90);
this.textBox3.Name = "textBox3";
this.textBox3.Size = new System.Drawing.Size(132, 20);
this.textBox3.TabIndex = 23;
this.textBox3.Text = "dd/mm/aaaa";
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(11, 90);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(63, 13);
this.label5.TabIndex = 34;
this.label5.Text = "Data Inicial:";
this.label5.Click += new System.EventHandler(this.label5_Click);

```

```

//
// textBox2
//
this.textBox2.Location = new System.Drawing.Point(93, 125);
this.textBox2.Name = "textBox2";
this.textBox2.Size = new System.Drawing.Size(132, 20);
this.textBox2.TabIndex = 24;
this.textBox2.Text = " dd/mm/aaaa";
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(11, 62);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(72, 13);
this.label4.TabIndex = 33;
this.label4.Text = "Equipamento:";
//
// lvDados
//
this.lvDados.AllowColumnReorder = true;
this.lvDados.AllowDrop = true;
this.lvDados.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.lvDados.BackColor = System.Drawing.SystemColors.ScrollBar;
this.lvDados.Font = new System.Drawing.Font("Microsoft Sans Serif",
9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)0));
this.lvDados.ForeColor = System.Drawing.Color.Black;
this.lvDados.FullRowSelect = true;
this.lvDados.GridLines = true;
this.lvDados.Location = new System.Drawing.Point(286, 104);
this.lvDados.Name = "lvDados";
this.lvDados.Size = new System.Drawing.Size(632, 462);
this.lvDados.TabIndex = 25;
this.lvDados.UseCompatibleStateImageBehavior = false;
this.lvDados.View = System.Windows.Forms.View.Details;
//
// tabPage2
//
this.tabPage2.Controls.Add(this.gerarGraficoCorrente);
this.tabPage2.Controls.Add(this.labelTeste);
this.tabPage2.Controls.Add(this.chart1);
this.tabPage2.Controls.Add(this.chart3);
this.tabPage2.Controls.Add(this.chart4);
this.tabPage2.Controls.Add(this.chart2);
this.tabPage2.Location = new System.Drawing.Point(4, 22);
this.tabPage2.Name = "tabPage2";
this.tabPage2.Padding = new System.Windows.Forms.Padding(3);
this.tabPage2.Size = new System.Drawing.Size(941, 767);
this.tabPage2.TabIndex = 1;
this.tabPage2.Text = "Gráficos de Corrente";
this.tabPage2.UseVisualStyleBackColor = true;
this.tabPage2.Click += new System.EventHandler(this.tabPage2_Click);
//
// gerarGraficoCorrente
//
this.gerarGraficoCorrente.Location = new System.Drawing.Point(25,
31);

```

```

this.gerarGraficoCorrente.Name = "gerarGraficoCorrente";
this.gerarGraficoCorrente.Size = new System.Drawing.Size(423, 23);
this.gerarGraficoCorrente.TabIndex = 16;
this.gerarGraficoCorrente.Text = "Gerar Gráficos de Corrente";
this.gerarGraficoCorrente.UseVisualStyleBackColor = true;
this.gerarGraficoCorrente.Click += new
System.EventHandler(this.gerarGraficoCorrente_Click);
//
// labelTeste
//
this.labelTeste.AutoSize = true;
this.labelTeste.Location = new System.Drawing.Point(448, 6);
this.labelTeste.Name = "labelTeste";
this.labelTeste.Size = new System.Drawing.Size(0, 13);
this.labelTeste.TabIndex = 15;
//
// chart1
//
this.chart1.BackColor = System.Drawing.Color.Gainsboro;
this.chart1.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.DiagonalRight;
this.chart1.BackImageTransparentColor =
System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224))))));
this.chart1.BackSecondaryColor = System.Drawing.Color.White;
this.chart1.BorderColor = System.Drawing.Color.Black;
chartArea13.AxisX.ScaleView.SmallScrollMinSize = 2D;
chartArea13.AxisX.ScrollBar.Size = 6D;
chartArea13.AxisX.Title = "Medições";
chartArea13.AxisY.Title = "Corrente (A)";
chartArea13.BackColor = System.Drawing.Color.Gainsboro;
chartArea13.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.VerticalCenter;
chartArea13.BackHatchStyle =
System.Windows.Forms.DataVisualization.Charting.ChartHatchStyle.Cross;
chartArea13.CursorX.IsUserEnabled = true;
chartArea13.CursorX.IsUserSelectionEnabled = true;
chartArea13.Name = "ChartArea1";
this.chart1.ChartAreas.Add(chartArea13);
legendCell12.Alignment = System.Drawing.ContentAlignment.TopLeft;
legendCell12.Name = "I(A)";
legendItem12.Cells.Add(legendCell12);
legend13.CustomItems.Add(legendItem12);
legend13.Name = "Legend1";
this.chart1.Legends.Add(legend13);
this.chart1.Location = new System.Drawing.Point(25, 96);
this.chart1.Name = "chart1";
series17.ChartArea = "ChartArea1";
series17.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
series17.Legend = "Legend1";
series17.Name = "Corrente IA";
series18.ChartArea = "ChartArea1";
series18.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
series18.Legend = "Legend1";
series18.Name = "Corrente IB";
series19.ChartArea = "ChartArea1";
series19.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
series19.Legend = "Legend1";
series19.Name = "Corrente IC";

```

```

        this.chart1.Series.Add(series17);
        this.chart1.Series.Add(series18);
        this.chart1.Series.Add(series19);
        this.chart1.Size = new System.Drawing.Size(423, 263);
        this.chart1.TabIndex = 14;
        this.chart1.Text = "chart1";
        //
        // chart3
        //
        this.chart3.Anchor =
        ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
        System.Windows.Forms.AnchorStyles.Left)));
        this.chart3.BackColor = System.Drawing.Color.Gainsboro;
        this.chart3.BackGradientStyle =
        System.Windows.Forms.DataVisualization.Charting.GradientStyle.DiagonalRight;
        this.chart3.BackImageTransparentColor =
        System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
        ((int)(((byte)(224))))));
        this.chart3.BackSecondaryColor = System.Drawing.Color.White;
        this.chart3.BorderColor = System.Drawing.Color.Black;
        chartArea14.AxisX.ScaleView.SmallScrollMinSize = 2D;
        chartArea14.AxisX.ScrollBar.Size = 6D;
        chartArea14.AxisX.Title = "Medições";
        chartArea14.AxisY.Title = "Corrente (A)";
        chartArea14.BackColor = System.Drawing.Color.Gainsboro;
        chartArea14.BackGradientStyle =
        System.Windows.Forms.DataVisualization.Charting.GradientStyle.VerticalCenter;
        chartArea14.BackHatchStyle =
        System.Windows.Forms.DataVisualization.Charting.ChartHatchStyle.Cross;
        chartArea14.CursorX.IsUserEnabled = true;
        chartArea14.CursorX.IsUserSelectionEnabled = true;
        chartArea14.Name = "ChartArea1";
        this.chart3.ChartAreas.Add(chartArea14);
        legendCell13.Alignment = System.Drawing.ContentAlignment.TopLeft;
        legendCell13.Name = "I(A)";
        legendItem13.Cells.Add(legendCell13);
        legend14.CustomItems.Add(legendItem13);
        legend14.Name = "Legend1";
        this.chart3.Legends.Add(legend14);
        this.chart3.Location = new System.Drawing.Point(482, 377);
        this.chart3.Name = "chart3";
        series20.ChartArea = "ChartArea1";
        series20.ChartType =
        System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series20.Legend = "Legend1";
        series20.Name = "Corrente IC";
        this.chart3.Series.Add(series20);
        this.chart3.Size = new System.Drawing.Size(423, 263);
        this.chart3.TabIndex = 13;
        this.chart3.Text = "chart3";
        //
        // chart4
        //
        this.chart4.Anchor =
        ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
        System.Windows.Forms.AnchorStyles.Left)));
        this.chart4.BackColor = System.Drawing.Color.Gainsboro;
        this.chart4.BackGradientStyle =
        System.Windows.Forms.DataVisualization.Charting.GradientStyle.DiagonalRight;
        this.chart4.BackImageTransparentColor =
        System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
        ((int)(((byte)(224))))));

```

```

        this.chart4.BackSecondaryColor = System.Drawing.Color.White;
        this.chart4.BorderlineColor = System.Drawing.Color.Black;
        chartArea15.AxisX.ScaleView.SmallScrollMinSize = 2D;
        chartArea15.AxisX.ScrollBar.Size = 6D;
        chartArea15.AxisX.Title = "Medições";
        chartArea15.AxisY.Title = "Corrente (A)";
        chartArea15.BackColor = System.Drawing.Color.Gainsboro;
        chartArea15.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.VerticalCenter;
        chartArea15.BackHatchStyle =
System.Windows.Forms.DataVisualization.Charting.ChartHatchStyle.Cross;
        chartArea15.CursorX.IsUserEnabled = true;
        chartArea15.CursorX.IsUserSelectionEnabled = true;
        chartArea15.Name = "ChartArea1";
        this.chart4.ChartAreas.Add(chartArea15);
        legendCell14.Alignment = System.Drawing.ContentAlignment.TopLeft;
        legendCell14.Name = "I(A)";
        legendItem14.Cells.Add(legendCell14);
        legend15.CustomItems.Add(legendItem14);
        legend15.Name = "Legend1";
        this.chart4.Legends.Add(legend15);
        this.chart4.Location = new System.Drawing.Point(25, 377);
        this.chart4.Name = "chart4";
        series21.ChartArea = "ChartArea1";
        series21.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series21.Legend = "Legend1";
        series21.Name = "Corrente IB";
        this.chart4.Series.Add(series21);
        this.chart4.Size = new System.Drawing.Size(423, 263);
        this.chart4.TabIndex = 12;
        this.chart4.Text = "chart4";
        //
        // chart2
        //
        this.chart2.BackColor = System.Drawing.Color.Gainsboro;
        this.chart2.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.DiagonalRight;
        this.chart2.BackImageTransparentColor =
System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224))))));
        this.chart2.BackSecondaryColor = System.Drawing.Color.White;
        this.chart2.BorderlineColor = System.Drawing.Color.Black;
        chartArea16.AxisX.ScaleView.SmallScrollMinSize = 2D;
        chartArea16.AxisX.ScrollBar.Size = 6D;
        chartArea16.AxisX.Title = "Medições";
        chartArea16.AxisY.Title = "Corrente (A)";
        chartArea16.BackColor = System.Drawing.Color.Gainsboro;
        chartArea16.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.VerticalCenter;
        chartArea16.BackHatchStyle =
System.Windows.Forms.DataVisualization.Charting.ChartHatchStyle.Cross;
        chartArea16.CursorX.IsUserEnabled = true;
        chartArea16.CursorX.IsUserSelectionEnabled = true;
        chartArea16.Name = "ChartArea1";
        this.chart2.ChartAreas.Add(chartArea16);
        legend16.Name = "Legend2";
        this.chart2.Legends.Add(legend16);
        this.chart2.Location = new System.Drawing.Point(482, 96);
        this.chart2.Name = "chart2";
        series22.ChartArea = "ChartArea1";

```

```

        series22.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series22.Legend = "Legend2";
        series22.Name = "Corrente IA";
        this.chart2.Series.Add(series22);
        this.chart2.Size = new System.Drawing.Size(423, 263);
        this.chart2.TabIndex = 11;
        this.chart2.Text = "chart2";
        //
        // tabPage3
        //
        this.tabPage3.Controls.Add(this.chart6);
        this.tabPage3.Controls.Add(this.chart7);
        this.tabPage3.Controls.Add(this.chart8);
        this.tabPage3.Controls.Add(this.gerarGraficoTensao);
        this.tabPage3.Controls.Add(this.chart5);
        this.tabPage3.Location = new System.Drawing.Point(4, 22);
        this.tabPage3.Name = "tabPage3";
        this.tabPage3.Padding = new System.Windows.Forms.Padding(3);
        this.tabPage3.Size = new System.Drawing.Size(941, 767);
        this.tabPage3.TabIndex = 2;
        this.tabPage3.Text = "Gráficos de Tensão";
        this.tabPage3.UseVisualStyleBackColor = true;
        //
        // chart6
        //
        this.chart6.BackColor = System.Drawing.Color.Gainsboro;
        this.chart6.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.DiagonalRight;
        this.chart6.BackImageTransparentColor =
System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224))))));
        this.chart6.BackSecondaryColor = System.Drawing.Color.White;
        this.chart6.BorderColor = System.Drawing.Color.Black;
        chartArea17.AxisX.ScaleView.SmallScrollMinSize = 2D;
        chartArea17.AxisX.ScrollBar.Size = 6D;
        chartArea17.AxisX.Title = "Medições";
        chartArea17.AxisY.Title = "Tensão (KV)";
        chartArea17.BackColor = System.Drawing.Color.Gainsboro;
        chartArea17.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.VerticalCenter;
        chartArea17.BackHatchStyle =
System.Windows.Forms.DataVisualization.Charting.ChartHatchStyle.Cross;
        chartArea17.CursorX.IsUserEnabled = true;
        chartArea17.CursorX.IsUserSelectionEnabled = true;
        chartArea17.Name = "ChartArea1";
        this.chart6.ChartAreas.Add(chartArea17);
        legendCell15.Alignment = System.Drawing.ContentAlignment.TopLeft;
        legendCell15.Name = "I(A)";
        legendItem15.Cells.Add(legendCell15);
        legend17.CustomItems.Add(legendItem15);
        legend17.Name = "Legend1";
        this.chart6.Legends.Add(legend17);
        this.chart6.Location = new System.Drawing.Point(479, 382);
        this.chart6.Name = "chart6";
        series23.ChartArea = "ChartArea1";
        series23.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series23.Legend = "Legend1";
        series23.Name = "Tensão VC";
        this.chart6.Series.Add(series23);
        this.chart6.Size = new System.Drawing.Size(423, 263);

```

```

        this.chart6.TabIndex = 24;
        this.chart6.Text = "chart8";
        //
        // chart7
        //
        this.chart7.BackColor = System.Drawing.Color.Gainsboro;
        this.chart7.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.DiagonalRight;
        this.chart7.BackImageTransparentColor =
System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224))))));
        this.chart7.BackSecondaryColor = System.Drawing.Color.White;
        this.chart7.BorderColor = System.Drawing.Color.Black;
        chartArea18.AxisX.ScaleView.SmallScrollMinSize = 2D;
        chartArea18.AxisX.ScrollBar.Size = 6D;
        chartArea18.AxisX.Title = "Medições";
        chartArea18.AxisY.Title = "Tensão (KV)";
        chartArea18.BackColor = System.Drawing.Color.Gainsboro;
        chartArea18.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.VerticalCenter;
        chartArea18.BackHatchStyle =
System.Windows.Forms.DataVisualization.Charting.ChartHatchStyle.Cross;
        chartArea18.CursorX.IsUserEnabled = true;
        chartArea18.CursorX.IsUserSelectionEnabled = true;
        chartArea18.Name = "ChartArea1";
        this.chart7.ChartAreas.Add(chartArea18);
        legendCell16.Alignment = System.Drawing.ContentAlignment.TopLeft;
        legendCell16.Name = "I(A)";
        legendItem16.Cells.Add(legendCell16);
        legend18.CustomItems.Add(legendItem16);
        legend18.Name = "Legend1";
        this.chart7.Legends.Add(legend18);
        this.chart7.Location = new System.Drawing.Point(30, 382);
        this.chart7.Name = "chart7";
        series24.ChartArea = "ChartArea1";
        series24.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series24.Legend = "Legend1";
        series24.Name = "Tensão VB";
        this.chart7.Series.Add(series24);
        this.chart7.Size = new System.Drawing.Size(423, 263);
        this.chart7.TabIndex = 23;
        this.chart7.Text = "chart7";
        //
        // chart8
        //
        this.chart8.BackColor = System.Drawing.Color.Gainsboro;
        this.chart8.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.DiagonalRight;
        this.chart8.BackImageTransparentColor =
System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224))))));
        this.chart8.BackSecondaryColor = System.Drawing.Color.White;
        this.chart8.BorderColor = System.Drawing.Color.Black;
        chartArea19.AxisX.ScaleView.SmallScrollMinSize = 2D;
        chartArea19.AxisX.ScrollBar.Size = 6D;
        chartArea19.AxisX.Title = "Medições";
        chartArea19.AxisY.Title = "Tensão (KV)";
        chartArea19.BackColor = System.Drawing.Color.Gainsboro;
        chartArea19.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.VerticalCenter;

```

```

        chartArea19.BackHatchStyle =
System.Windows.Forms.DataVisualization.Charting.ChartHatchStyle.Cross;
        chartArea19.CursorX.IsUserEnabled = true;
        chartArea19.CursorX.IsUserSelectionEnabled = true;
        chartArea19.Name = "ChartArea1";
        this.chart8.ChartAreas.Add(chartArea19);
        legendCell17.Alignment = System.Drawing.ContentAlignment.TopLeft;
        legendCell17.Name = "I(A)";
        legendItem17.Cells.Add(legendCell17);
        legend19.CustomItems.Add(legendItem17);
        legend19.Name = "Legend1";
        this.chart8.Legends.Add(legend19);
        this.chart8.Location = new System.Drawing.Point(479, 93);
        this.chart8.Name = "chart8";
        series25.ChartArea = "ChartArea1";
        series25.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series25.Legend = "Legend1";
        series25.Name = "Tensão VA";
        this.chart8.Series.Add(series25);
        this.chart8.Size = new System.Drawing.Size(423, 263);
        this.chart8.TabIndex = 22;
        this.chart8.Text = "chart6";
        //
        // gerarGraficoTensao
        //
        this.gerarGraficoTensao.Location = new System.Drawing.Point(30, 28);
        this.gerarGraficoTensao.Name = "gerarGraficoTensao";
        this.gerarGraficoTensao.Size = new System.Drawing.Size(423, 23);
        this.gerarGraficoTensao.TabIndex = 21;
        this.gerarGraficoTensao.Text = "Gerar Gráficos de Tensões";
        this.gerarGraficoTensao.UseVisualStyleBackColor = true;
        this.gerarGraficoTensao.Click += new
System.EventHandler(this.gerarGraficoTensao_Click);
        //
        // chart5
        //
        this.chart5.BackColor = System.Drawing.Color.Gainsboro;
        this.chart5.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.DiagonalRight;
        this.chart5.BackImageTransparentColor =
System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224))))));
        this.chart5.BackSecondaryColor = System.Drawing.Color.White;
        this.chart5.BorderlineColor = System.Drawing.Color.Black;
        chartArea20.AxisX.ScaleView.SmallScrollMinSize = 20;
        chartArea20.AxisX.ScrollBar.Size = 60;
        chartArea20.AxisX.Title = "Medições";
        chartArea20.AxisY.Title = "Tensão (KV)";
        chartArea20.BackColor = System.Drawing.Color.Gainsboro;
        chartArea20.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.VerticalCenter;
        chartArea20.BackHatchStyle =
System.Windows.Forms.DataVisualization.Charting.ChartHatchStyle.Cross;
        chartArea20.CursorX.IsUserEnabled = true;
        chartArea20.CursorX.IsUserSelectionEnabled = true;
        chartArea20.Name = "ChartArea1";
        this.chart5.ChartAreas.Add(chartArea20);
        legendCell18.Alignment = System.Drawing.ContentAlignment.TopLeft;
        legendCell18.Name = "I(A)";
        legendItem18.Cells.Add(legendCell18);
        legend20.CustomItems.Add(legendItem18);

```

```

        legend20.Name = "Legend1";
        this.chart5.Legends.Add(legend20);
        this.chart5.Location = new System.Drawing.Point(30, 93);
        this.chart5.Name = "chart5";
        series26.ChartArea = "ChartArea1";
        series26.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series26.Legend = "Legend1";
        series26.Name = "Tensão VA";
        series27.ChartArea = "ChartArea1";
        series27.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series27.Legend = "Legend1";
        series27.Name = "Tensão VB";
        series28.ChartArea = "ChartArea1";
        series28.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series28.Legend = "Legend1";
        series28.Name = "Tensão VC";
        this.chart5.Series.Add(series26);
        this.chart5.Series.Add(series27);
        this.chart5.Series.Add(series28);
        this.chart5.Size = new System.Drawing.Size(423, 263);
        this.chart5.TabIndex = 20;
        this.chart5.Text = "chart5";
        //
        // tabPage4
        //
        this.tabPage4.Controls.Add(this.chart9);
        this.tabPage4.Controls.Add(this.gerarGraficoPotencia);
        this.tabPage4.Controls.Add(this.chart10);
        this.tabPage4.Location = new System.Drawing.Point(4, 22);
        this.tabPage4.Name = "tabPage4";
        this.tabPage4.Padding = new System.Windows.Forms.Padding(3);
        this.tabPage4.Size = new System.Drawing.Size(941, 767);
        this.tabPage4.TabIndex = 3;
        this.tabPage4.Text = "Gráficos de Potência";
        this.tabPage4.UseVisualStyleBackColor = true;
        //
        // chart9
        //
        this.chart9.BackColor = System.Drawing.Color.Gainsboro;
        this.chart9.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.DiagonalRight;
        this.chart9.BackImageTransparentColor =
System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224))))));
        this.chart9.BackSecondaryColor = System.Drawing.Color.White;
        this.chart9.BorderlineColor = System.Drawing.Color.Black;
        chartArea11.AxisX.ScaleView.SmallScrollMinSize = 2D;
        chartArea11.AxisX.ScrollBar.Size = 6D;
        chartArea11.AxisX.Title = "Medições";
        chartArea11.AxisY.Title = "Potência Reativa(KVar)";
        chartArea11.BackColor = System.Drawing.Color.Gainsboro;
        chartArea11.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.VerticalCenter;
        chartArea11.BackHatchStyle =
System.Windows.Forms.DataVisualization.Charting.ChartHatchStyle.Cross;
        chartArea11.CursorX.IsUserEnabled = true;
        chartArea11.CursorX.IsUserSelectionEnabled = true;
        chartArea11.Name = "ChartArea1";
        this.chart9.ChartAreas.Add(chartArea11);

```

```

        legendCell10.Alignment = System.Drawing.ContentAlignment.TopLeft;
        legendCell10.Name = "I(A)";
        legendItem10.Cells.Add(legendCell10);
        legend11.CustomItems.Add(legendItem10);
        legend11.Name = "Legend1";
        this.chart9.Legends.Add(legend11);
        this.chart9.Location = new System.Drawing.Point(476, 89);
        this.chart9.Name = "chart9";
        series15.ChartArea = "ChartArea1";
        series15.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series15.Legend = "Legend1";
        series15.Name = "Potência Reativa";
        this.chart9.Series.Add(series15);
        this.chart9.Size = new System.Drawing.Size(423, 263);
        this.chart9.TabIndex = 25;
        this.chart9.Text = "chart6";
        //
        // gerarGraficoPotencia
        //
        this.gerarGraficoPotencia.Location = new System.Drawing.Point(27,
24);
        this.gerarGraficoPotencia.Name = "gerarGraficoPotencia";
        this.gerarGraficoPotencia.Size = new System.Drawing.Size(423, 23);
        this.gerarGraficoPotencia.TabIndex = 24;
        this.gerarGraficoPotencia.Text = "Gerar Gráficos de Potências";
        this.gerarGraficoPotencia.UseVisualStyleBackColor = true;
        this.gerarGraficoPotencia.Click += new
System.EventHandler(this.gerarGraficoPotencia_Click);
        //
        // chart10
        //
        this.chart10.BackColor = System.Drawing.Color.Gainsboro;
        this.chart10.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.DiagonalRight;
        this.chart10.BackImageTransparentColor =
System.Drawing.Color.FromArgb(((int)(((byte)224))), ((int)(((byte)224))),
((int)(((byte)224))));
        this.chart10.BackSecondaryColor = System.Drawing.Color.White;
        this.chart10.BorderlineColor = System.Drawing.Color.Black;
        chartArea12.AxisX.ScaleView.SmallScrollMinSize = 20;
        chartArea12.AxisX.ScrollBar.Size = 60;
        chartArea12.AxisX.Title = "Medições";
        chartArea12.AxisY.Title = "Potência Ativa (KW)";
        chartArea12.BackColor = System.Drawing.Color.Gainsboro;
        chartArea12.BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.VerticalCenter;
        chartArea12.BackHatchStyle =
System.Windows.Forms.DataVisualization.Charting.ChartHatchStyle.Cross;
        chartArea12.CursorX.IsUserEnabled = true;
        chartArea12.CursorX.IsUserSelectionEnabled = true;
        chartArea12.Name = "ChartArea1";
        this.chart10.ChartAreas.Add(chartArea12);
        legendCell11.Alignment = System.Drawing.ContentAlignment.TopLeft;
        legendCell11.Name = "I(A)";
        legendItem11.Cells.Add(legendCell11);
        legend12.CustomItems.Add(legendItem11);
        legend12.Name = "Legend1";
        this.chart10.Legends.Add(legend12);
        this.chart10.Location = new System.Drawing.Point(27, 89);
        this.chart10.Name = "chart10";
        series16.ChartArea = "ChartArea1";

```

```

        series16.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        series16.Legend = "Legend1";
        series16.Name = "Potência Ativa";
        this.chart10.Series.Add(series16);
        this.chart10.Size = new System.Drawing.Size(423, 263);
        this.chart10.TabIndex = 23;
        this.chart10.Text = "chart10";
        //
        // pictureBox1
        //
        this.pictureBox1.ErrorImage =
((System.Drawing.Image)(resources.GetObject("pictureBox1.ErrorImage")));
        this.pictureBox1.InitialImage =
((System.Drawing.Image)(resources.GetObject("pictureBox1.InitialImage")));
        this.pictureBox1.Location = new System.Drawing.Point(823, 0);
        this.pictureBox1.Name = "pictureBox1";
        this.pictureBox1.Size = new System.Drawing.Size(111, 28);
        this.pictureBox1.TabIndex = 1;
        this.pictureBox1.TabStop = false;
        //
        // Form1
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(957, 727);
        this.Controls.Add(this.pictureBox1);
        this.Controls.Add(this.tabControl1);
        this.Icon =
((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
        this.Name = "Form1";
        this.Text = "Consulta Equipamentos Automatizados VTS por Renato
Souto";
        this.Load += new System.EventHandler(this.Form1_Load);
        this.tabControl1.ResumeLayout(false);
        this.tabPage1.ResumeLayout(false);
        this.panel3.ResumeLayout(false);
        this.panel3.PerformLayout();
        this.panel2.ResumeLayout(false);
        this.panel2.PerformLayout();
        this.panel1.ResumeLayout(false);
        this.panel1.PerformLayout();
        this.tabPage2.ResumeLayout(false);
        this.tabPage2.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)(this.chart1)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.chart3)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.chart4)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.chart2)).EndInit();
        this.tabPage3.ResumeLayout(false);
        ((System.ComponentModel.ISupportInitialize)(this.chart6)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.chart7)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.chart8)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.chart5)).EndInit();
        this.tabPage4.ResumeLayout(false);
        ((System.ComponentModel.ISupportInitialize)(this.chart9)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.chart10)).EndInit();

        ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
        this.ResumeLayout(false);
    }

```

```
#endregion
```

```
private System.Windows.Forms.TabControl tabControl1;  
private System.Windows.Forms.TabPage tabPage2;  
private System.Windows.Forms.TabPage tabPage3;  
private System.Windows.Forms.TabPage tabPage4;  
private System.Windows.Forms.DataVisualization.Charting.Chart chart3;  
private System.Windows.Forms.DataVisualization.Charting.Chart chart4;  
private System.Windows.Forms.DataVisualization.Charting.Chart chart2;  
private System.Windows.Forms.DataVisualization.Charting.Chart chart1;  
private System.Windows.Forms.Label labelTeste;  
private System.Windows.Forms.Button gerarGraficoCorrente;  
private System.Windows.Forms.DataVisualization.Charting.Chart chart6;  
private System.Windows.Forms.DataVisualization.Charting.Chart chart7;  
private System.Windows.Forms.DataVisualization.Charting.Chart chart8;  
private System.Windows.Forms.Button gerarGraficoTensao;  
private System.Windows.Forms.DataVisualization.Charting.Chart chart5;  
private System.Windows.Forms.DataVisualization.Charting.Chart chart9;  
private System.Windows.Forms.Button gerarGraficoPotencia;  
private System.Windows.Forms.DataVisualization.Charting.Chart chart10;  
private System.Windows.Forms.TabPage tabPage1;  
private System.Windows.Forms.Panel panel3;  
private System.Windows.Forms.TextBox txtBancoDados;  
private System.Windows.Forms.Button btnSelecionaBD;  
private System.Windows.Forms.Panel panel2;  
private System.Windows.Forms.Label label11;  
private System.Windows.Forms.Label label10;  
private System.Windows.Forms.Label label8;  
private System.Windows.Forms.TextBox textBox4;  
private System.Windows.Forms.Label label7;  
private System.Windows.Forms.ComboBox comboBox1;  
private System.Windows.Forms.TextBox valorMinimo;  
private System.Windows.Forms.Label label3;  
private System.Windows.Forms.Label label2;  
private System.Windows.Forms.TextBox valorMaximo;  
private System.Windows.Forms.Panel panel1;  
private System.Windows.Forms.Label label9;  
private System.Windows.Forms.Button buttonOk;  
private System.Windows.Forms.TextBox buscaEquipamento;  
private System.Windows.Forms.Label label6;  
private System.Windows.Forms.TextBox textBox3;  
private System.Windows.Forms.Label label5;  
private System.Windows.Forms.TextBox textBox2;  
private System.Windows.Forms.Label label4;  
private System.Windows.Forms.ListView lvDados;  
private System.Windows.Forms.Label label1;  
private System.Windows.Forms.PictureBox pictureBox1;
```

```
}
```

```
}
```