



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**  
**CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA**  
**UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA**

**TRABALHO DE CONCLUSÃO DE CURSO**

**Desenvolvimento da IHM de um sistema de monitoramento de descargas  
parciais no ambiente de uma subestação de sistemas elétricos**

---

**Rafael de Melo Silva Santos**

**Aluno**

---

**Maria de Fátima Queiroz Vieira**

**Professora Orientadora**

**CAMPINA GRANDE**  
**JUNHO DE 2016**

RAFAEL DE MELO SILVA SANTOS

**Desenvolvimento da IHM de um sistema de monitoramento de descargas  
parciais no ambiente de uma subestação de sistemas elétricos**

*Relatório de Trabalho de Conclusão de  
Curso submetido à Unidade Acadêmica de 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.*

Áreas de Concentração: Automação e Controle

Orientador (a):  
Professora Maria de Fátima Queiroz Vieira, PhD.

Campina Grande, Paraíba  
Junho de 2016

RAFAEL DE MELO SILVA SANTOS

**Desenvolvimento da IHM de um sistema de monitoramento de descargas  
parciais no ambiente de uma subestação de sistemas elétricos**

*Relatório de Trabalho de Conclusão de  
Curso submetido à Unidade Acadêmica de 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.*

Áreas de Concentração: Automação e Controle

Aprovado em

**Professor**

Universidade Federal de Campina Grande  
Avaliador

**Professora Maria de Fátima Queiroz Vieira, PhD**

Universidade Federal de Campina Grande  
Orientador(a), UFCG

## **AGRADECIMENTOS**

Inicialmente, gostaria de agradecer a Deus por ter me dado a força necessária durante esse período em que abdiquei de muitos momentos com minha família e amigos. Aos meus pais, Sebastião e Sandra, e minha irmã, Gabriela, por me apoiar em todos os momentos de minha vida. À minha noiva, Ana Karoliny, por ter sido tão paciente e compreensiva durante toda a graduação.

Agradeço também a professora Maria de Fátima Queiroz Vieira pela excelente orientação no processo de realização do TCC na UFCG, e por todo o empenho em me mostrar o melhor caminho para realização das atividades propostas, além de ter sido uma excelente professora durante as disciplinas de graduação.

Agradeço também a meus amigos, em especial Claviano e Cleverson, que sempre me apoiaram mesmo à distância, e a Alex, Luciano, Rodrigo, Eduardo, e tantos outros que me proporcionaram bons momentos ao longo desta caminhada.

A todos estes, meu mais sincero agradecimento.

## RESUMO

Este trabalho de conclusão de curso teve como objetivo projetar a Interface Homem-Máquina (IHM) que integrará o sistema de supervisão de descargas parciais em equipamentos de uma subestação do sistema elétrico. Este projeto faz parte de um projeto maior desenvolvido em cooperação entre as Universidades de Huddersfield e de Strathclyde no Reino Unido (UK) e a UFCG, a partir de um acordo de cooperação. Neste projeto da IHM foi adotado o Método de Concepção de Interfaces Ergonômicas (MCIE) desenvolvido no Laboratório de Interfaces Homem-Máquina (LIHM) da Universidade Federal de Campina Grande (UFCG) e apresentado na disciplina Informática Industrial, do Curso de Engenharia Elétrica da UFCG. O projeto da IHM teve como objetivo oferecer recursos aos usuários para: configurar o ambiente de monitoramento em um software supervisor; facilitar a entrada de dados oriundos de um conjunto de radiômetros e, exibir o estado dos equipamentos monitorados através de radiômetros, desenvolvidos pela equipe do UK, os quais são utilizados como sensores de descargas parciais. Os dados dos radiômetros utilizados na construção e teste da IHM foram fornecidos pela equipe do Reino Unido, como uma amostra de dados em um arquivo Excel, resultado de experimentos com os radiômetros. Estes dados, adquiridos no campo, são processados no Matlab/Simulink, com um algoritmo desenvolvido pela equipe do UK, para fornecer a intensidade e localização estimada das descargas parciais nos equipamentos. A IHM foi desenvolvida para o supervisor Indusoft Web Studio, utilizando a ferramenta IWS. A comunicação entre o supervisor e o Matlab foi realizada através do protocolo OPC (OLE for Process Control). Os dados referentes às medições de cada radiômetro são armazenados no Access. A IHM desenvolvida permite configurar o ambiente de monitoramento e a partir do acesso aos dados, exibe os níveis de descarga parciais no ambiente monitorado, emitindo alarmes e a localização provável quando estes níveis ultrapassam limites de segurança. A interface desenvolvida foi testada a partir de um conjunto de dados disponibilizados pela equipe do Reino Unido.

## **ABSTRACT**

This course work consisted in designing the human-machine interface (HMI) which integrates the partial discharge monitoring system to be used in substations of electrical systems. This project is part of a larger project developed in cooperation between the Universities of Huddersfield and Strathclyde in the United Kingdom (UK) and UFCG in Brazil. For the HMI development it was adopted the Design Method for Ergonomic Interfaces (MCIE), which was developed in the Human-Machine Interfaces Laboratory (LIHM) at the Federal University of Campina Grande (UFCG), and was taught in the discipline Industrial Informatics at the Electrical Engineering Undergraduate Course at UFCG. The HMI project aims to provide the users with resources for: configuring the monitoring environment in a supervisory software; facilitating data input which originates from a set of radiometers and, to display the status of devices monitored by radiometers. The radiometers were developed by the UK team, and are used as partial discharge sensors. The sample data from radiometers used in the construction and testing of HMI was provided by the UK team, in an Excel file, and resulted from experiments with radiometers. These data, acquired during field tests is processed in Matlab / Simulink using an algorithm developed by the UK team, in order to provide the intensity and estimated location of partial discharges inside the electric equipment in a substation. The HMI component was developed using the IWS tool associated to the InduSoft Web Studio supervisory system. Communication between the supervisory system and Matlab was performed using the OPC protocol (OLE for Process Control). Data acquired from the radiometers are stored in Access files. The HMI developed allows to configure the monitoring environment based on the incoming data, generating screens to display the discharge levels in the monitored environment; sending alarms and calculating the probable location where levels exceed safe limits. The HMI was tested using the sample data provided by the UK team.

# Sumário

LISTA DE FIGURAS.....	8
LISTA DE SIGLAS.....	9
1. Introdução.....	10
1.1. Objetivos.....	12
1.2. Método.....	12
1.3. Apresentação do Texto.....	13
2. Projeto de Interface Homem-Máquina e Representação de Dados Complexos na Engenharia ..	15
2.1. Interfaces de equipamentos para monitoramento de descargas parciais.....	15
2.2. Representação de Dados Complexos na IHM.....	17
2.3. Princípios Gerais de Projeto.....	19
2.4. Descrição dos Dados do Projeto.....	20
3. Levantamento de Requisitos e Especificação da IHM.....	25
3.1. Método MCIE.....	25
3.2. Representação UML.....	29
3.2.1. Diagrama de caso de uso.....	29
3.2.2. Diagrama de sequência.....	30
3.3. Lista FAST.....	31
3.4. Objetivos de Usabilidade (em ordem de prioridade).....	31
4. Desenvolvimento da IHM.....	33
4.1. Ferramentas adotadas: Indusoft Web Studio e Matlab.....	33
4.2. Desenvolvimento das Telas.....	34
4.3. Conexão da IHM - Definição dos Tags no IWS.....	47
4.4. Uso de Animações na IHM.....	51
4.5. Definição de Alarmes.....	52
4.6. Desenvolvimento de Scripts.....	54
4.7. Comunicação Matlab/Simulink e Indusoft.....	55
5. Resultados e Considerações.....	60
5.1. Proposta de Testes de Integração.....	61
5.2. Considerações Finais.....	61
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	63
ANEXO - CÓDIGOS VBScript e SQL.....	65

## LISTA DE FIGURAS

Figura 1: Interface e equipamentos de monitoramento periódico e contínuo .....	16
Figura 2: Representação dos radiômetros .....	18
Figura 3: Tela exibindo os dados do radiômetro 3.....	22
Figura 4: Fluxo de dados entre o IWS e o BD .....	23
Figura 5: Fluxograma representando a rotina de cálculo dos parâmetros dos radiômetros .....	24
Figura 6: Etapas MCIE [6] .....	26
Figura 7: Modelo da tarefa.....	28
Figura 8: Diagrama de caso de uso .....	29
Figura 9: Diagrama de sequência - Cenário de rotina .....	30
Figura 10: Diagrama de sequência - Cenário de exceção.....	30
Figura 11: Modelo da Tarefa (High Level) .....	34
Figura 12: Modelo de Tarefa Atualizado .....	35
Figura 13: Hierarquia de telas .....	35
Figura 14: Tela com nome da IHM e botões de navegação entre telas.....	35
Figura 15: Tela inicial.....	36
Figura 16: Monitor PD .....	37
Figura 17: Dados atuais e disposição dos radiômetros (Current Data).....	38
Figura 18: Janela Pop-up com dados dos radiômetros .....	39
Figura 19: Dados históricos de um radiômetro.....	40
Figura 20: Tabela contendo todos os dados dos radiômetros.....	40
Figura 21: Dados históricos dos radiômetros.....	41
Figura 22: Tela de configuração do sistema.....	42
Figura 23: Tela para inserção da localização dos radiômetros .....	43
Figura 24: Tela para calibração do sistema .....	44
Figura 25: Tela Update Data para calibração do sistema.....	45
Figura 26: Tela Location View para inserção e remoção dos radiômetros .....	46
Figura 27: Categorias de TAGs .....	48
Figura 28: Tags do projeto.....	50
Figura 29:Classe RAD (Atributos dos radiômetros) .....	50
Figura 30: Classe Substation (Atributos da subestação) .....	50
Figura 31: Tela de alarmes .....	53
Figura 32: Tela de alarmes recentes .....	54
Figura 33: Tela de histórico de alarmes .....	54
Figura 34: Fluxo de dados entre Matlab/Simulink e IWS.....	57
Figura 35: Configuração da comunicação via OPC .....	58
Figura 36: Leitura e escrita de Tags do IWS .....	59
Figura 37: Teste de conexão entre o IWS e o BD .....	60

## LISTA DE SIGLAS

<b>IHM</b>	Interface Homem-Máquina
<b>IWS</b>	Indusoft Web Studio
<b>SCADA</b>	Supervisory Control and Data Acquisition
<b>MCIE</b>	Método de Concepção de Interfaces Ergonômicas
<b>SQL</b>	Structured Query Language

## 1. Introdução

Este projeto de conclusão de curso consiste no desenvolvimento da IHM para um sistema de monitoramento de descargas parciais internas, em equipamentos instalados no pátio de uma subestação de um sistema elétrico. A IHM foi desenvolvida para o sistema de supervisão (SCADA), utilizando o Indusoft Web Studio [6]. O principal objetivo da IHM é possibilitar a configuração do sistema e a representação de um grande volume de dados de monitoramento para apoiar as decisões do operador; antecipando e prevenindo falhas, e facilitando a manutenção e substituição de equipamentos, de acordo com o conceito das *smartgrids*.

Neste projeto, a representação clara do grande volume de dados fornecidos por cada radiômetro é o problema principal. Nele foi buscada uma representação que simplificasse a interpretação dos dados e reduzisse o tempo de decisão do operador. Este projeto foi desenvolvido em parceria entre os departamentos de Engenharia Elétrica das Universidades: *Strathclyde e Huddersfield*, no Reino Unido e a UFCG no Brasil. No Reino Unido, o pesquisador responsável é o Professor Ian Glover, que coordena as pesquisas relativas ao desenvolvimento e integração dos sensores e na UFCG, a professora Maria de Fatima Queiroz Vieira que coordena os trabalhos relacionados ao desenvolvimento do projeto da IHM do sistema de monitoramento.

### Descargas parciais

Quando uma tensão elétrica é aplicada aos terminais de um equipamento que possui algum tipo de isolamento elétrico entre as partes energizadas (ar, SF<sub>6</sub>, resinas, vidros, entre outros), devido à falhas no material podem ocorrer descargas em algumas áreas do dielétrico. Estas descargas poderão ocorrer de duas formas: quando a intensidade do campo elétrico for maior no local da descarga, e na segunda forma, quando a constante dielétrica do material for menor em algumas áreas. A partir destas descargas, podem ocorrer: luminosidade; pulsos eletromagnéticos conduzidos e radiados; ruído acústico; aumento de temperatura e reações químicas localizadas [4].

Dentre os fenômenos descritos destacam-se as descargas parciais internas aos equipamentos que operam em alta tensão. Estas descargas ocorrem no interior do material isolante, em cavidades ou inclusões (corpos estranhos incorporados no material). As cavidades podem suportar um valor mais baixo de tensão aplicada visto que a constante dielétrica do gás

nestas cavidades é menor que a constante dielétrica do material isolante. Assim, a distribuição de tensão será maior nas cavidades, forçando a ocorrência das descargas parciais.

As descargas parciais descritas causam a deterioração do material isolante do equipamento, podendo acarretar no mau funcionamento, e conseqüentemente, comprometendo todo o sistema elétrico. Logo, é de extrema importância o conhecimento da localização das descargas parciais para que uma medida preventiva possa ser tomada, evitando perdas materiais e de carga. Para que a medida preventiva seja colocada em prática, é necessário que a localização da ocorrência das descargas seja conhecida dos operadores da instalação.

#### O projeto do sistema de monitoramento das descargas parciais

O projeto consiste no desenvolvimento de uma rede de sensores (radiômetros); um software para o monitoramento dos dados dotado de uma Interface Homem-Máquina (IHM) e de algoritmos capazes de processar os dados e fornecer a localização das descargas. Os sensores, ou radiômetros foram desenvolvidos pela equipe da Universidade de Huddersfield e pela UFCG, para detectar a radiação emitida pelas descargas.

Na IHM projetada, o usuário deverá ser capaz de configurar o sistema durante a instalação, periodicamente calibrar os sensores (radiômetros) e, monitorar a ocorrência das descargas parciais aplicando algoritmos sobre os dados coletados, de modo a estimar a localização das descargas, associando-a à intensidade da descarga, entre outros parâmetros. O usuário poderá também ajustar o intervalo de coleta dos dados pelos radiômetros.

#### A Interface Homem Máquina (IHM) e o software SCADA

O desenvolvimento da IHM foi feito utilizando as ferramentas oferecidas pelo ambiente SCADA - Indusoft Web Studio, comercializado pela Schneider Electric, utilizando o método MCIE, desenvolvido no LIHM da UFCG [5]. O SCADA da Indusoft foi escolhido por oferecer todos os recursos necessários, além de haver uma licença educacional disponível para uso nos projetos do Laboratório de Interface Homem-Máquina (LIHM-UFCG).

Como os dados coletados por cada radiômetro foram armazenados em um arquivo .xls (Excel) pela equipe da universidade parceira responsável pela aquisição dos dados, e sabendo que o IWS não pode realizar manipulações com os dados estando neste formato, então realizamos a conversão manual do arquivo .xls para o arquivo .mdb (Access), permitindo assim a manipulação dos dados pelo IWS.

## 1.1. Objetivos

O projeto tem os objetivos listados a seguir:

- Desenvolvimento da IHM do sistema de monitoramento de descargas parciais, de acordo com os requisitos propostos pelas equipes das universidades participantes do projeto.
- Representação de um grande volume de dados de forma a simplificar a compreensão dos dados e reduzir o tempo de análise e decisão do operador.
- Armazenamento dos dados oriundos dos radiômetros e dados de configuração
- Permitir a apresentação de dados enviados pelos sensores e permitir a apresentação dos resultados gerados por algoritmos. Facilitar o aprendizado no uso da IHM e minimizar a taxa de erros nas decisões tomadas pelo operador.

## 1.2. Método

O procedimento adotado neste projeto segue o método de criação de interfaces ergonômicas – MCIE, desenvolvido no LIHM e, apresentado na disciplina Informática Industrial. Em relação ao armazenamento dos dados coletados pelos radiômetros, foi realizada uma simples conversão do arquivo recebido no formato .xls (Excel) para o formato .mdb (Access). Esta conversão foi feita manualmente abrindo o arquivo excel e em *Salvar como...* escolhemos o formato .mdb. Essa conversão foi necessária para compatibilizar o arquivo com o IWS e possibilitar a manipulação dos dados do arquivo a partir da linguagem SQL, visto que o IWS possibilita apenas a exibição dos dados em um arquivo .xls (Excel) e impossibilita sua manipulação. A especificação e projeto de um BD que armazene todos os arquivos .mdb, que serão continuamente atualizados, serão realizados em uma etapa futura deste projeto.

As especificações da IHM foram discutidas entre as equipes das universidades parceiras, e a especificação proposta pela professora Maria de Fátima resultou de discussões com a equipe da Universidade de Huddersfield.

O projeto foi organizado em etapas: na primeira etapa, foi realizada uma revisão bibliográfica sobre o problema das descargas parciais e sobre as interfaces dos equipamentos utilizados para monitorar este fenômeno. Neste tópico foi estudado como as descargas são

geradas, os efeitos que podem causar a um equipamento, como detectá-las, entre outros tópicos abordados no projeto.

Também foram estudadas as representações deste tipo de dados tanto em equipamentos comercializados quanto na literatura técnica científica. Foram também estudadas as especificações propostas para a IHM a ser desenvolvida. O principal objetivo foi entender a natureza dos dados e representá-los adequadamente.

Na etapa seguinte foi estudado o software Indusoft Web Studio, usando como fonte a documentação disponível e, vídeos tutoriais disponibilizados pelo fornecedor.

Dando continuidade, na terceira etapa, a especificação original da IHM foi revisada e codificada. Finalmente foram realizados testes de requisitos e redigida a documentação do projeto.

Segue a lista de etapas do trabalho:

- **Etapa 1:** Revisão bibliográfica;
- **Etapa 2:** Treinamento do software Indusoft Web Studio;
- **Etapa 3:** Desenvolvimento da IHM;
- **Etapa 4:** Testes e Redação da Documentação do Projeto;
- **Etapa 5:** Redação do Relatório do projeto do Trabalho de Conclusão de Curso (TCC)

### **1.3. Apresentação do Texto**

Este trabalho está organizado da seguinte forma: na seção 2 será apresentada a revisão bibliográfica sobre o projeto de Interfaces Homem-Máquina, com enfoque na representação de dados complexos na engenharia, destacando os tipos de representações utilizadas e a representação escolhida para este projeto. Por fim, será apresentada a descrição dos dados recebidos no formato .xls (Excel), e como foram tratados para serem exibidos aos usuários da IHM.

Na seção 3 serão apresentadas as especificações e requisitos do projeto, resultantes da discussão entre as universidades cooperantes: UFCG, Huddersfield e Strathclyde. Na seção 4 serão apresentadas as etapas do desenvolvimento da IHM. Por fim, na seção 5, serão apresentados os resultados dos testes da comunicação: (a) entre o arquivo .mdb (Access) e a

interface e, (b) entre o Matlab e Indusoft; e por fim, a conexão entre as telas e objetos de base do Indusoft. Nesta seção são também sugeridos testes de integração, os quais serão realizados pelas equipes do Reino Unido, Strathclyde quando concluírem o sistema de monitoramento. E para finalizar na seção 5, são tecidas considerações acerca do projeto desenvolvido.

## **2. Projeto de Interface Homem-Máquina e Representação de Dados Complexos na Engenharia**

Neste capítulo são apresentados os princípios do projeto de uma interface homem-máquina centrada no usuário, destacando conceitos como a ergonomia da interface, usabilidade, e o Método de Concepção de Interfaces Ergonômicas (MCIE) , destacando suas etapas de desenvolvimento. Nela também são abordados os problemas comuns ao projeto de interfaces para representação de grandes volumes de dados em ambientes industriais, dando ênfase a representação adotada para este projeto.

### **2.1. Interfaces de equipamentos para monitoramento de descargas parciais**

As maiores vantagens do monitoramento contínuo, em comparação com análises periódicas, são a automatização das medições, integração de parâmetros com o sistema de controle da planta via Modbus e a possibilidade de ativar alarmes remotos indicando a necessidade de investigações mais detalhadas dos dados, por exemplo. Os equipamentos de monitoramento podem arquivar os dados e eventos associados ao registro de data e hora (*timestamp*).[14]

Existem diversos equipamentos no mercado que podem realizar o monitoramento contínuo e/ou periódico das descargas parciais, os quais são comercializados com ou sem uma IHM. Dentre os equipamentos disponíveis no mercado brasileiro destaca-se o MONTRANO[16] produzido pela Adimarco, que é um sistema de monitoramento contínuo com uma interface Web para acesso aos dados. Outros equipamentos da Adimarco são o OMS 605 e o OMS 800, que podem realizar o monitoramento periódico das descargas parciais.

Para o monitoramento contínuo das descargas parciais, além do MONTRANO comercializado pela Adimarco, há também o IMA-DP desenvolvido pela Eletrobras/ Cepel [15], e o GuardII produzido pela EuroVolt [14]. O objetivo de introduzir estes equipamentos foi ilustrar as características das interfaces concebidas para visualizar os recursos disponíveis para o monitoramento de descargas parciais.

Em geral, quando analisa os dados exibidos por sistemas de monitoramento, o usuário tende a preferir a visualização de imagens ao invés de textos, principalmente quando estes dados forem volumosos [17].

De acordo com a literatura [17], [18], a forma mais eficiente de representação de dados em um ambiente industrial e aquela induz o menor número de erros é baseada em elementos visuais tais como gráficos ou outros objetos que forneçam as informações necessárias ao usuário sem que ele precise realizar um grande esforço para compreensão.

Embora a representação visual gráfica induza maior produtividade, muitas IHM's ainda são projetadas para apresentar os dados de forma textual (ex.: tabelas). Esta representação exige maior esforço de memorização e compreensão e pode ocasionar erros de interpretação dos dados.

Outra forma de representar os dados é a representação a partir de objetos 3D [18]. Esta representação seria ideal para este projeto, visto que a propagação das ondas eletromagnéticas ocorre nos três eixos do plano cartesiano, mas devido à complexidade de implementação, no prazo do projeto, optou-se pela representação visual com objetos 2D, plano (x,y). A figura 1 contém os equipamentos citados acima, Montrano (Figura 1.D), OMS800 (Figura 1.B), OMS605 (Figura 1.C) e GuardII (Figura 1.A).



**Figura 1:** Interface e equipamentos de monitoramento periódico e contínuo

## 2.2. Representação de Dados Complexos na IHM

O nível de automação adotado em plantas industriais pode variar bastante. Os sistemas mais simples dependem do trabalho de operadores próximos à planta industrial para, por exemplo, transportar, montar ou separar peças, monitorar variáveis através de mostradores, operar painéis de controle e outros equipamentos no chão de fábrica. Por outro lado, sistemas mais sofisticados permitem que operadores possam monitorar e controlar o processo remotamente através de sistemas supervisórios, que coletam os dados da planta industrial, formatam esses dados e os apresentam em terminais localizados em salas de controle.

Em sistemas de grande e médio porte, o conjunto de informações monitoradas pelo operador, a partir do supervisório, é tipicamente elevado e a introdução de um sistema supervisório gera vantagens e benefícios como: facilidade de acesso à informação; redução de custos; integração com sistemas EMS (Enterprise Management Systems); aumento da qualidade; maior segurança e aumento da capacidade produtiva. Assim, a tendência no ambiente industrial tem sido a instalação de sistemas de supervisão e a automatização do trabalho manual, especialmente em tarefas repetitivas e sistemáticas. No entanto, o operador pode ser requisitado a interferir no processo em situações críticas ou de exceção. Nestes casos, é comum que a tomada de decisões por parte do operador necessite da análise de uma grande quantidade de dados e variáveis, em múltiplas dimensões, tais como: temperatura, pressão, vazão, nível, corrente elétrica, potência consumida, etc.

Sabe-se que operadores industriais trabalham em equipe, compartilham informações, agem muitas vezes de forma coordenada e possuem prazos restritivos para tomada de decisões. Esse contexto precisa ser considerado no projeto de um sistema supervisório, buscando minimizar a probabilidade de sobrecarga cognitiva dos operadores e a ocorrência de erros com consequências potencialmente graves. Neste cenário, uma das principais questões é: como apoiar operadores de sistemas supervisórios na tomada de decisões que dependem da análise de um grande volume de dados com naturezas distintas?

O projeto das IHMs de supervisórios é constituído pelas seguintes etapas: estudo do processo a ser automatizado; coleta de dados e representação das variáveis (tags); planejamento de alarmes; planejamento da hierarquia de telas; concepção da interface; conexão com um repositório de dados; conexão com CLP's (Controlador Lógico Programável) e UTR's

(Unidade Terminal Remota); definição do console de acesso; integração e testes; e o treinamento dos operadores do sistema.

Conhecendo as necessidades de representação de dados nas aplicações de engenharia, tais como a representação em tabelas, gráficos de barras, representação 3D, entre outras, neste projeto foi escolhida a representação 2D, como ilustrado na figura 2. Nesta representação, os dados dos radiômetros são representados como circunferências, cujos raios variam de acordo com a potência medida. Ou seja, se a incidência de descargas parciais estiver aumentando em um equipamento situado próximo a um radiômetro, o raio da representação aumenta proporcionalmente à soma das potências medidas nas três fases do sistema, de acordo com a equação 1.



**Figura 2:** Representação dos radiômetros

$$Raio = P1 + P2 + P3 \quad (1)$$

$$SetorX = \frac{Px}{P1+P2+P3} \quad (2)$$

Os setores dos círculos que representam os radiômetros, representam as potências medidas em cada fase do sistema; as quais são representadas nas cores: vermelho, verde e azul. A área dos setores é calculada de acordo com a eq. 2 e, representa proporcionalmente a potência em cada uma das fases. A representação visual da potência medida pelos radiômetros em círculos com setores e áreas proporcionais à potência medida e com uma borda colorida representando limites de potência, foi proposta pela professora Maria de Fátima Vieira, na

UFCEG. Enquanto o cálculo do raio e da área dos setores foi definido pelo Professor Ian Glover. Esta representação visual dos dados medidos pelos radiômetros foi proposta visando facilitar a análise do volume dos dados e sua complexidade na tomada de decisão pelo operador do sistema.

### 2.3. Princípios Gerais de Projeto

De acordo com as notas de aula da disciplina de Informática Industrial [6], ministrada pela Professora Maria de Fátima, alguns princípios devem ser respeitados antes do projeto de uma interface gráfica ser iniciado. O cumprimento desses princípios fará com que a IHM seja bem estruturada, dentro das especificações do cliente, e proporcionará uma melhor experiência de uso aos usuários. O projeto deve ser concebido do ponto de vista do que é melhor para o usuário, ao invés do que é mais rápido e fácil de implementar. Os princípios são os seguintes:

- Reconhecimento das diversas situações de uso;
- Caracterização dos usuários e situações de forma precisa e completa;
- Manter a consistência: Terminologia, *layout*, sequência de ações;
- Fornecer atalhos para usuários experientes;
  - A experiência cresce de acordo com a frequência de utilização, então é preciso diminuir o número de iterações para aumentar a velocidade durante a utilização.
- Fornecer realimentação informativa;
  - Toda ação do usuário deve corresponder a uma resposta do sistema. Respostas simples para ações frequentes, e respostas elaboradas para ações complexas.
- Projetar diálogos que levem a um fechamento;
  - Sequências de ações devem ser organizadas em grupos, com princípio, meio, fim e realimentação do sistema.
- Fornecer procedimentos simples para recuperação de erros;
- Projetar robustez para evitar erros graves;
- Permitir a reversão de ações;
- Reduzir a carga na memória do usuário.

A supervisão de um ambiente industrial é uma aplicação crítica que demanda o compromisso entre qualidade, produtividade e segurança no trabalho. O equilíbrio entre esses fatores passa pela definição de uma interface com o usuário [6]. Desta forma, para que as ações realizadas pelo usuário não sejam comprometidas por falta de clareza ou informações, a interface deve seguir as recomendações que serão citadas. O não cumprimento pode causar a navegação difícil e ineficiente, telas sobrecarregadas, projeto visual que conduz à fadiga, entre outros fatores.

- Recomendações para o projeto de um sistema supervisório [6]:
  - Conheça o usuário e a tarefa que realiza
  - Organize a informação em agrupamentos por tópico, por tema,
  - Adote uma representação simples do processo industrial
  - Minimize a carga da memória de curta duração, evite a memorização.
  - Desenvolva uma estrutura de telas de fácil navegação
  - Localize o usuário na estrutura de telas identificando cada tela da interface
  - A tela do sinóptico deve resumir a operação do sistema e conter links para sua operação
  - Identifique todos os elementos na tela do sinóptico
  - Projete a tela com foco claro, títulos que conduzem o leitor, layout compacto
  - Priorize os alarmes
  - Considere o nível de luminosidade no ambiente de supervisão
  - Destaque e diferencie adequadamente cada variável.

## **2.4. Descrição dos Dados do Projeto**

Neste tópico será descrita a forma com que o pacote de dados é enviado à interface para que seja representado adequadamente, informando detalhadamente o que cada variável contida no arquivo de dados significa. Por fim, um fluxograma é apresentado de maneira a facilitar a compreensão do fluxo de dados neste projeto.

Para o desenvolvimento do projeto foram utilizadas amostras de dados, coletadas por três radiômetros. O radiômetro é um sensor utilizado para medir a radiação associada aos pulsos eletromagnéticos emitidos por descargas parciais internas aos equipamentos. Os dados foram recebidos no formato .xls (arquivo do aplicativo Microsoft Excel). Os dados da amostra

obtida, para cada radiômetro, são apresentados na tabela 1. No ambiente real, no qual o sistema será instalado, os dados serão atualizados periodicamente na forma de arquivos *.mdb* (Microsoft Access).

Os dados representados na tabela 1 são: *Date & Time*, a data e a hora (*timestamp*) da medida do radiômetro, no formato DD/MM/AAAA e HH/MM; *MoteID* que identifica o radiômetro no qual foi realizada a medição; *Pulse#*, que fornece a quantidade de pulsos de tensão que foram medidos entre o início da medição e o *timestamp*; o campo *Time(ns)* que representa o intervalo de tempo da medição dos pulsos de tensão, representado em nano segundos. Por fim, os campos *Ch1*, *Ch2* e *Ch3*, fornecem os níveis de tensão, em miliVolts, em cada canal de medição, cada um associado a uma fase do sinal medido: *Ch1* para fase 1, *Ch2* para fase 2 e *Ch3* para fase 3. Nos testes realizados, foram utilizados apenas os dados do *Ch1*, dado que nos foi informado que os níveis de tensão representados nos canais *Ch2* e *Ch3*, correspondem a ruídos.

**Tabela 1:** Excerto da amostra de dados coletados

<b>Date &amp; Time</b>	<b>MoteID</b>	<b>Pulse#</b>	<b>Time (ns)</b>	<b>Ch1(mV)</b>	<b>Ch2(mV)</b>	<b>Ch3(mV)</b>
12/10/2015 16:33	3	1	2225940	310	0	595
12/10/2015 16:33	3	2	2237420	785	0	595
12/10/2015 16:33	3	3	2248920	1313	0	595
12/10/2015 16:33	3	4	2260560	0	0	601
12/10/2015 16:33	7	1	13013260	0	0	234
12/10/2015 16:33	7	2	22238120	524	0	225
12/10/2015 16:33	7	3	22249620	1137	0	225
12/10/2015 16:33	8	1	1353140	539	539	1351
12/10/2015 16:33	8	2	1359460	633	539	1351
12/10/2015 16:33	8	3	1365780	703	539	1351

Os dados obtidos no projeto da IHM foram fornecidos antes do processamento pelo algoritmo de localização das descargas parciais. Isto é, são os dados crus (*rawdata*) coletados pelos sensores e enviados para processamento na estação de monitoramento do usuário. O tratamento do *rawdata* para apresentação na tela de monitoramento foi realizado com base em um conjunto de equações fornecidas pelo pesquisador responsável pela equipe do Reino Unido, Professor Ian Glover. Estas equações são apresentadas a seguir.

Na figura 3 é exibida a tela na qual os dados referentes ao radiômetro 7, são apresentados aos usuários do sistema. Nela, os campos *P1*, *P2*, *P3*, *Pulse Count* e *Pulse Rate*,

correspondem aos dados exibidos na tabela 1, após o processamento, de acordo com as equações 3, 4, 5, 6 e 7, as quais são explicitadas a seguir.

Field	Value
RAD_ID	3
P1	199.962250
P2	0.000000
P3	228.456857
C	28
R	797.848999

**Figura 3:** Tela exibindo os dados do radiômetro 3

$$P1 = \frac{N1reset}{Tm} \quad (3)$$

$$P2 = \frac{N2reset}{Tm} \quad (4)$$

$$P3 = \frac{N3reset}{Tm} \quad (5)$$

$$C = Max(pulse) \quad (6)$$

$$R = \frac{Max(pulse)}{Tm} \quad (7)$$

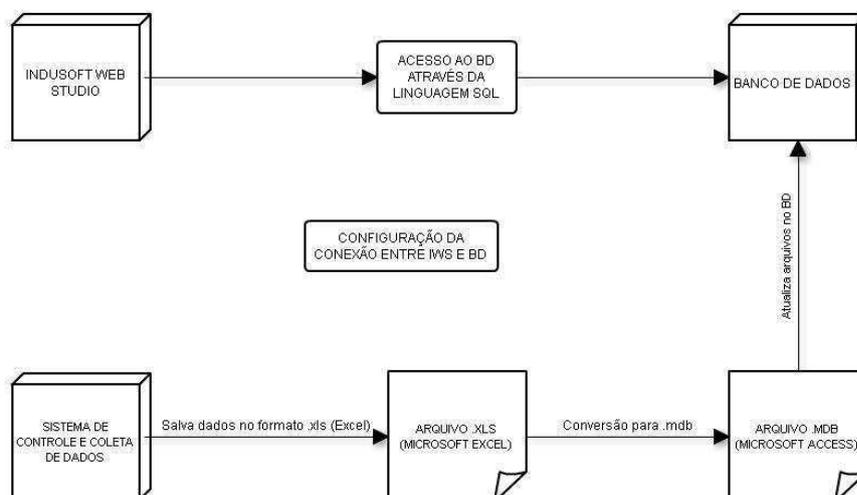
As equações 3, 4 e 5 representam as potências nas três fases do sinal sistema medidas pelo radiômetro. O cálculo consiste na razão entre o número de resets ( $N1reset$ ,  $N2reset$  e  $N3reset$ ) e o período de tempo da medição  $Tm$ . Os  $NXreset$  são obtidos a partir da contagem do número de vezes em que a tensão em um dos canais de medição dos radiômetros ( $Ch1$ ,  $Ch2$  e  $Ch3$ ) decresce.

Por exemplo, tomando como base os dados da coluna  $Ch1$  da tabela 1, pode-se verificar que para os pulsos 1, 2 e 3, os valores de tensão do  $Ch1$  são crescentes, assumindo os valores 310, 785 e 1313, respectivamente. Assim, quando o pulso 4 é medido, o valor 0 no  $Ch1$ , faz com que o número de resets seja igual a 1 ( $N1reset = 1$ ) pois, os valores lidos no  $Ch1$

reverteram e iniciaram uma queda, iniciando um novo ciclo de contagem. Desta forma, a contagem do número de resets continua até que o tempo  $T_m$  seja atingido. Para que o tratamento dos dados seja realizado, é necessário que o valor de  $T_m$  (*Time Measurement*) tenha sido definido na tela de calibração do sistema, que será mostrada na seção 4.2 (figura 25). O tempo  $T_m$  define o período de tempo entre o início das medições até que o valor de  $T_m$  seja atingido. Caso os valores de P1, P2 e P3 sejam maiores que 0; um fator de correção de 0,5 deve ser acrescentado a esse valor, que corresponde à estimativa de que após  $T_m$  segundos a tensão alcançou metade da tensão de limiar. Mas se P1, P2 ou P3 forem iguais a zero, não há necessidade desta correção nos seus valores.

Nas equações 6 e 7, o parâmetro  $Max(pulse)$  refere-se ao número máximo de pulsos medidos no período de tempo  $T_m$ . Nelas, C corresponde ao número máximo de pulsos e, R à razão entre este número C e o tempo de medição  $T_m$ . O processamento dos dados foi realizado executando as equações apresentadas programadas na linguagem VBscript [12].

Para o desenvolvimento da IHM, a equipe da universidade parceira responsável pela aquisição de dados forneceu um arquivo no formato .xls (Excel) contendo uma amostra dos dados para que testes pudessem ser realizados. Para que o IWS pudesse manipular os dados fornecidos, foi necessário realizar a conversão do arquivo .xls para .mdb. Esta conversão foi feita manualmente, mas foi decidido que a equipe responsável pela aquisição dos dados forneceria o arquivo no formato .mdb. O BD é atualizado automaticamente (em processo a ser definido em etapas futuras), onde os novos arquivos de dados .mdb são salvos diretamente no BD. O acesso do IWS ao BD foi realizado a partir da linguagem SQL. O fluxograma da figura 4 corresponde ao processo descrito anteriormente.

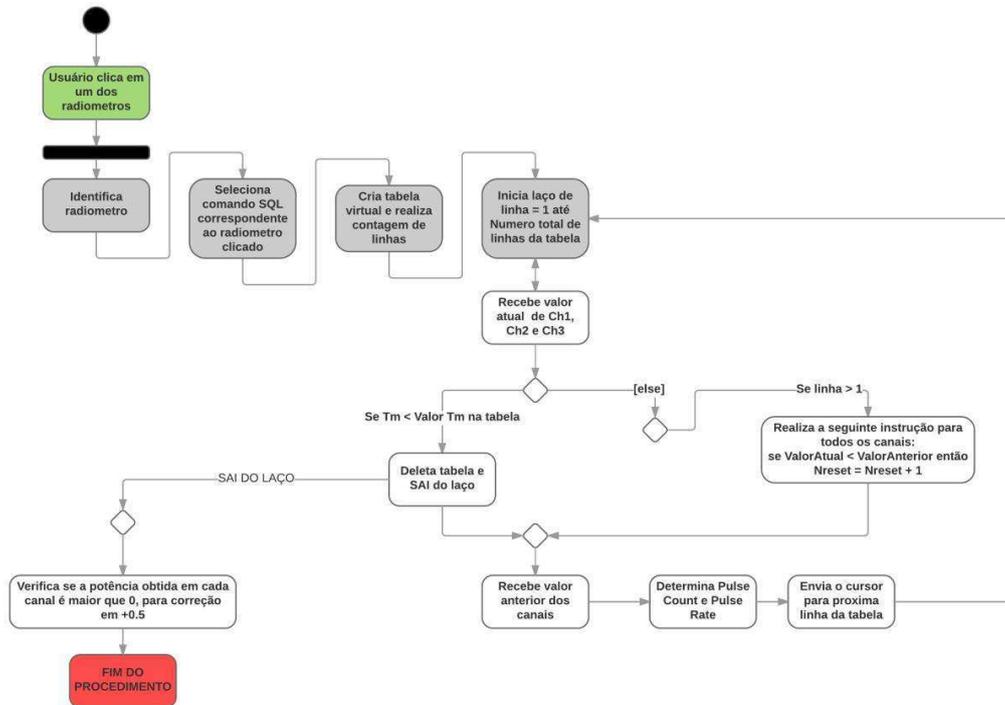


**Figura 4:** Fluxo de dados entre o IWS e o BD

O fluxograma que ilustra a rotina é mostrado na figura 5, enquanto o código desenvolvido é apresentado no Anexo (Determinação dos parâmetros de cada radiômetro).

DIAGRAMA 1

Rafael de Melo Silva Santos | January 28, 2016



**Figura 5:** Fluxograma representando a rotina de cálculo dos parâmetros dos radiômetros

### **3. Levantamento de Requisitos e Especificação da IHM**

Esta seção apresenta a concepção e desenvolvimento de uma interface gráfica, também denominada Interface Homem-Máquina (IHM), que integrará o software de um sistema supervisor empregado no monitoramento da incidência de descargas parciais internas aos equipamentos de uma subestação de um sistema elétrico. Esta interface permitirá ao operador do sistema localizar níveis anormais de descargas parciais em equipamentos da instalação. O método adotado neste processo é descrito a seguir.

#### **3.1. Método MCIE**

O Método para Concepção de Interfaces Ergonômicas, ou MCIE, é um método que apoia a concepção e o desenvolvimento de interfaces bem estruturadas que atendem as necessidades do usuário e do cliente. O método MCIE consiste das etapas descritas a seguir:

1. Levantamento das necessidades e estabelecimento de objetivos do cliente
2. Levantamento do Perfil dos usuários
3. Análise da Tarefa
4. Construção do modelo da interação
  - 4.1. Paradigma de interação
  - 4.2. Estilos de interação
  - 4.3. Projeto dos objetos e das ações
  - 4.4. Criação das metáforas e dos manipuladores
5. Projeto Visual
  - 5.1. Apresentação da informação
  - 5.2. Codificação da informação
6. Avaliação do Projeto
7. Monitoração do uso e realização de atualizações

A seguir são apresentados os produtos de cada etapa do MCIE, instanciados para este projeto. A figura 6 explica as etapas do MCIE. O método se inicia pela análise de requisitos, onde o desenvolvedor deve analisar o contexto no qual a IHM será inserida, as características

dos usuários que utilizaram regularmente a interface, e após ter conhecimento dessas especificações o desenvolvedor pode desenvolver o modelo da tarefa para o projeto, especificando detalhadamente cada tarefa que será realizada pela IHM.

Na etapa de modelagem da interação, o desenvolvedor deve elaborar uma lista contendo todos os objetos e ações da análise da tarefa para que possam ser realizadas a modelagem da apresentação, que seria um esboço do projeto visual da IHM, e a modelagem da navegação, que compreende como o usuário navegará entre as telas da IHM. Ao fim destas modelagens, deve-se realiza o protótipo da IHM e logo em seguida a avaliação desse protótipo. Caso melhorias devam ser realizadas, o desenvolvedor deve retornar a etapa que exija melhoria (Análise de requisitos, Modelagem da interação, Prototipagem).

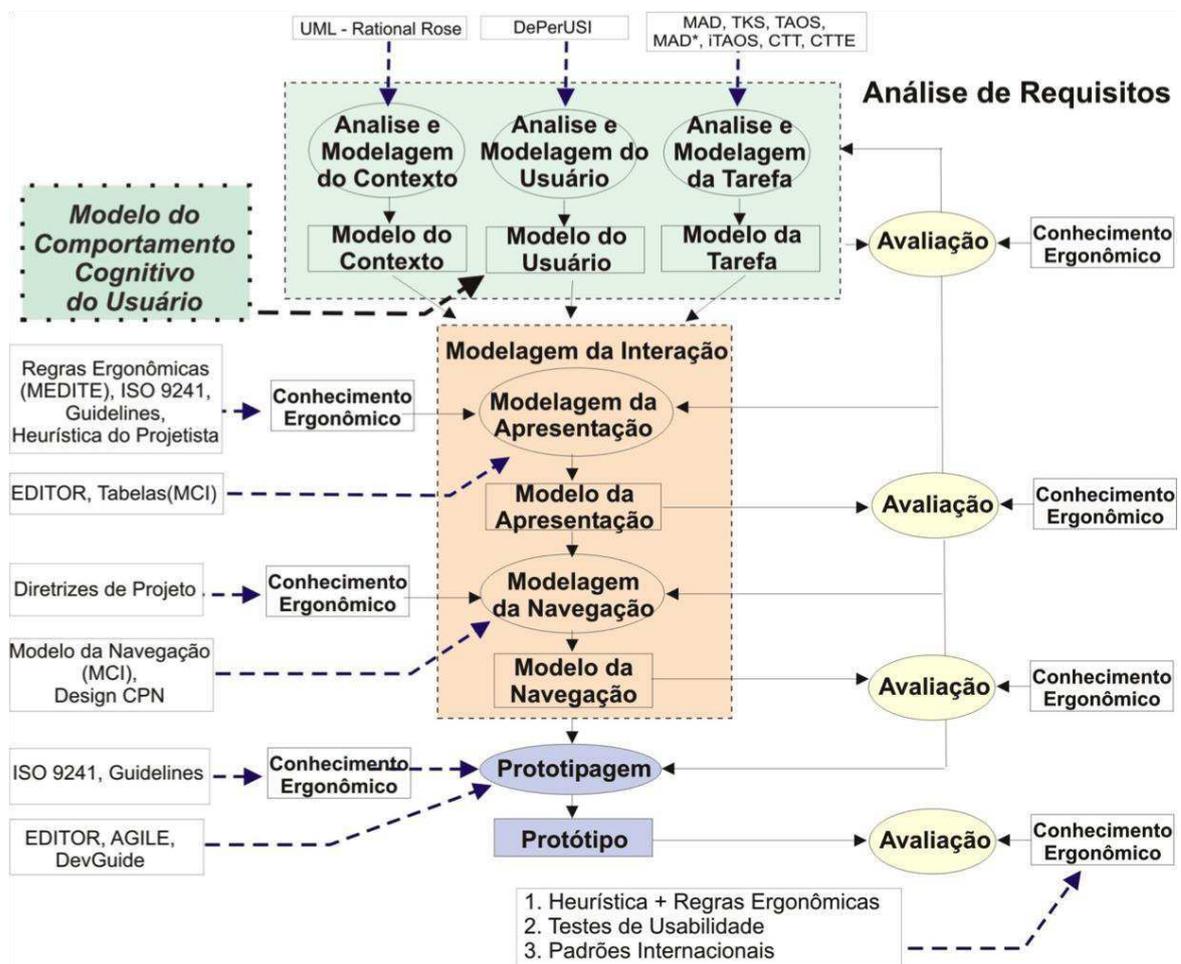


Figura 6: Etapas MCIE [6]

## Perfil do Usuário

Diz respeito as características do usuário que são escolhidas pelo projetista, de acordo com a relevância para o projeto. Neste caso, foram analisadas as características: idade, sexo, motivações, frequência de utilização, experiência com o sistema, conhecimento da tarefa realizada pela IHM, estilo cognitivo, habilidades motoras e de percepção.[6] No método MCIE, o levantamento do perfil do usuário pode ser apoiado por formulários, por observação, ou por questionário. Neste projeto foram identificadas as características a seguir:

- Idade: a partir de 20 anos;
- Sexo: sem predominância;
- Objetivos: Facilidade no monitoramento do sistema;
- Frequência de utilização: Alta;
- Nível de Experiência com o sistema após treinamento: Elevada;
- Aprendizado: por treinamento;
- Estilo cognitivo: Persistência (Elevada),
- Nível de percepção visual (Alta).

### **Análise e Modelagem da tarefa**

De acordo com o MCIE [6], envolve a compreensão e representação das ações e sequências de ações necessárias para realização da tarefa;identificando:

- Porque são necessárias;
- Qual o fluxo da informação;
- Qual a contribuição do usuário para o processo;
- Que pode ser automatizado com o objetivo de aumentar: a produtividade, a eficiência e a qualidade do sistema.

O modelo da tarefa proposto para este projeto fornece a descrição detalhada das tarefas, subtarefas e métodos envolvidos na utilização do sistema de monitoramento de descargas parciais utilizando o sistema supervisorio. Este modelo, utilizando o formalismo MAD, está representado na Figura 7.

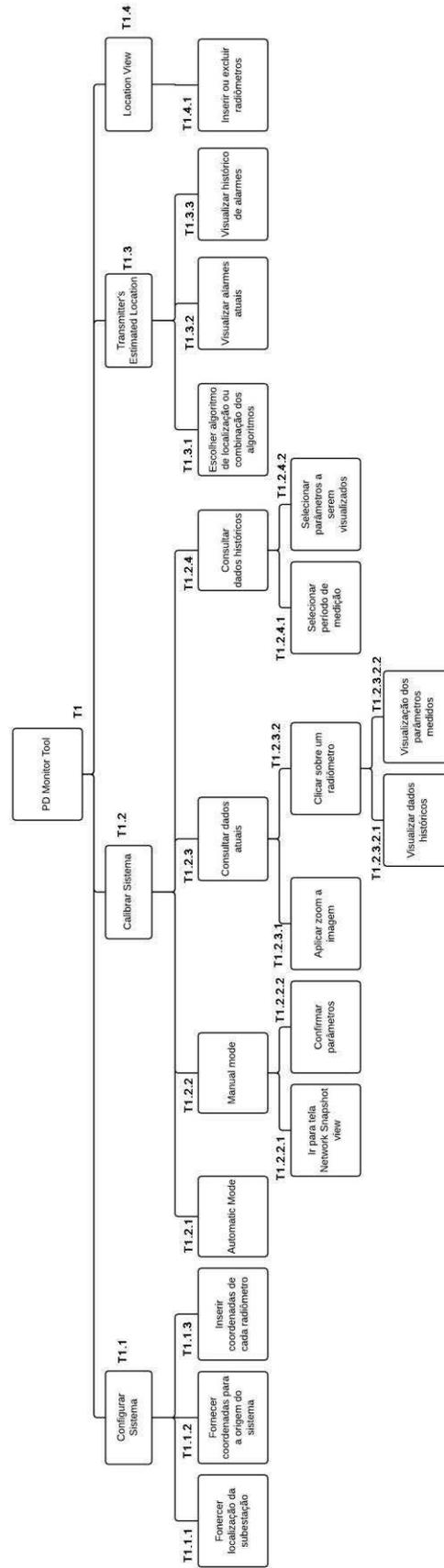


Figura 7: Modelo da tarefa

### 3.2. Representação UML

O UML (Unified Modeling Language) [6] é uma linguagem padrão para elaboração de modelos em projetos de software, e que possui diagramas que podem ser combinados com a finalidade de obter todas as visões e aspectos do sistema. Os principais objetivos dos diagramas UML são:

- Visualizar o sistema como ele é ou como deseja que ele seja.
- Especificar a estrutura ou o comportamento de um sistema.
- Proporcionar um guia para construção do sistema.
- Documentar o sistema.

#### 3.2.1. Diagrama de caso de uso

O diagrama de caso de uso é um diagrama dinâmico que modela o comportamento do sistema percebido por atores externos. Caracteriza um objetivo a ser alcançado pelo ator externo ao utilizar o sistema. No diagrama da figura 8, pode-se observar que qualquer usuário pode acessar as telas de *home*, *monitor pd*, *locate pd*, *alarm*, *report*, *system configure*, *system calibrate* e *location view*, mas as únicas telas que precisam acessar o BD são *monitor pd*, *locate pd* e *alarm*.

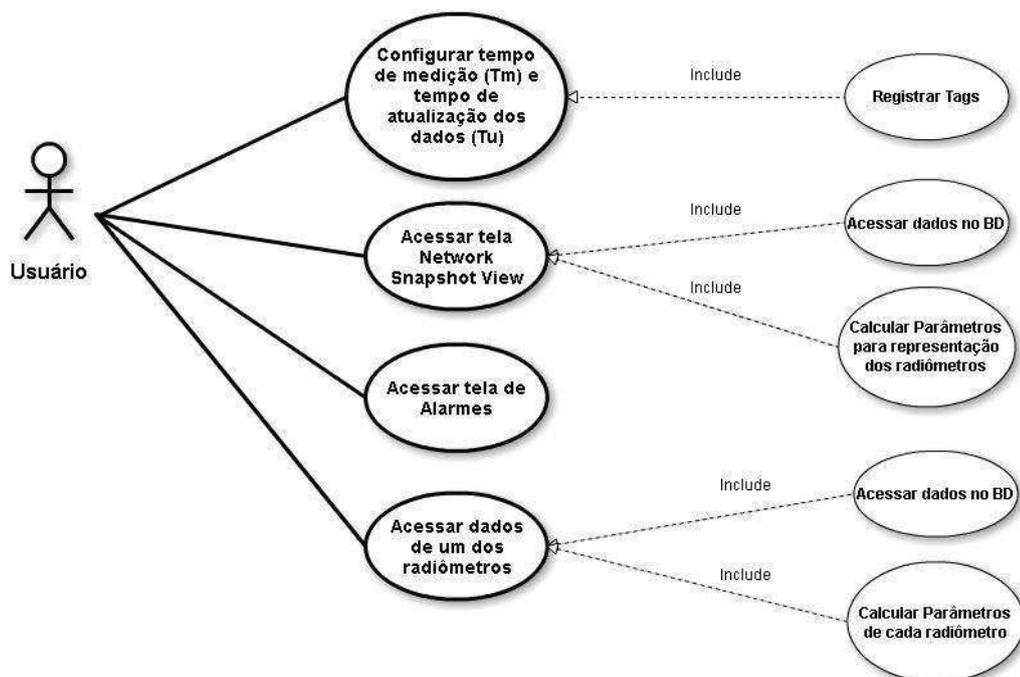


Figura 8: Diagrama de caso de uso

### 3.2.2. Diagrama de sequência

Os diagramas de sequência permitem a modelagem de processos através da troca de mensagens entre os objetos do sistema. Cada objeto é uma linha vertical, e as mensagens são setas que partem do objeto que invoca um outro objeto. Nos diagramas de sequência nas ilustrados nas figuras 8 e 9, são apresentados um Cenário de rotina (figura 9), no qual o usuário acessa a tela “Network Snapshot View” para verificar os dados medidos por cada radiômetro, e um Cenário de exceção (Figura 10), no qual um usuário tenta consultar os dados históricos referentes a um radiômetro em um período que não existem medições.

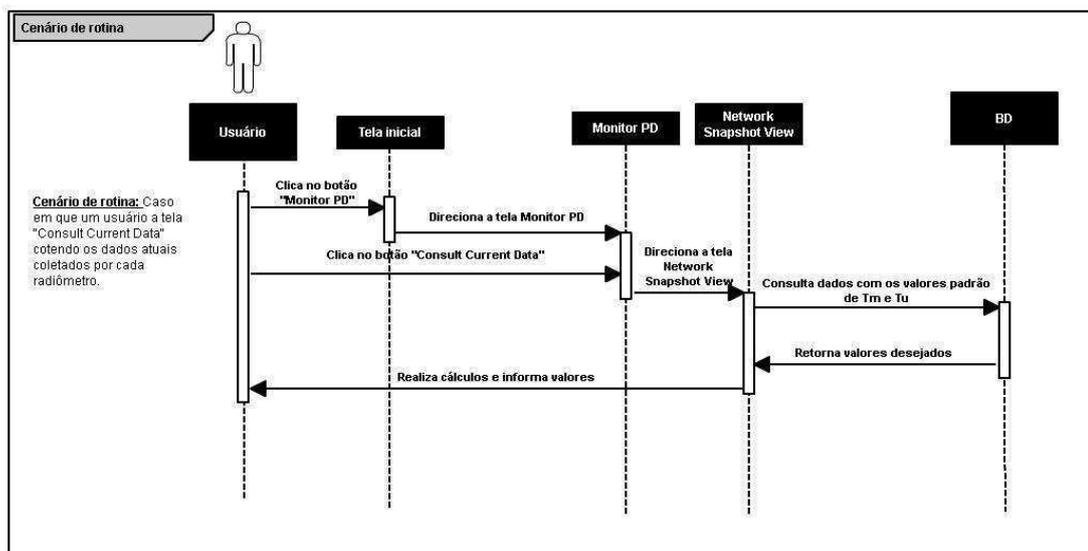


Figura 9: Diagrama de sequência - Cenário de rotina

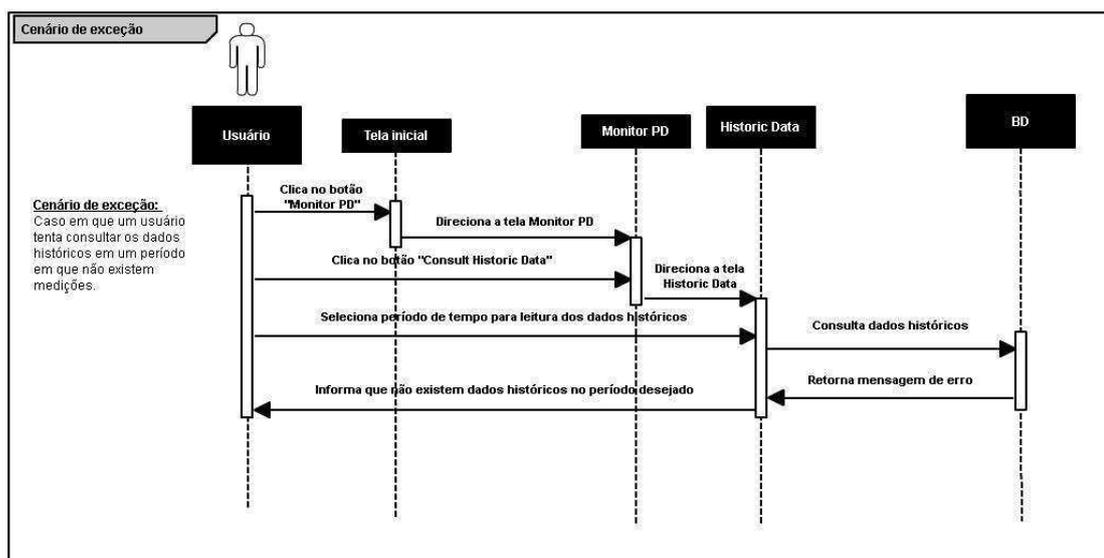


Figura 10: Diagrama de sequência - Cenário de exceção

### 3.3. Lista FAST

Apesar de não fazer parte do MCIE, a seguir é apresentada a lista FAST [13] com a especificação inicial do projeto da IHM, a qual é apresentada a seguir.

#### **Objetos que fazem parte do ambiente que circunda o sistema:**

- Radiômetros
- Equipamentos que compõem a subestação
- IHM

#### **Objetos produzidos pelo sistema:**

- Medição de tensão;
- Medição de potência;
- Elaboração de curvas de históricos de medições;
- Disparo de alarmes;

#### **Objetos usados para o sistema produzir suas funções:**

- Monitor 15”
- Sensores (Radiômetros);
- Alarmes;

#### **Lista de operações que manipulam ou interagem com os objetos:**

- Monitoração de sensores e atuadores;
- Monitoração do estado dos radiômetros;
- Acionamento dos alarmes;
- Localização aproximada das descargas parciais;

#### **Lista de restrições:**

- Velocidade de processamento do computador no qual a IHM será instalada;
- Velocidade do sistema de coleta de dados;
- Transferência de dados entre os dispositivos e o BD.

### 3.4. Objetivos de Usabilidade (em ordem de prioridade)

- Facilitar o monitoramento dos equipamentos da subestação;

- Fornecer ao usuário uma visão completa sobre o sistema, ajudando-o a avaliar de maneira rápida e precisa o estado dos equipamentos com relação as descargas parciais geradas naquela área;
- Apoiar decisões do usuário através de uma interface clara e intuitiva.

No próximo capítulo serão apresentadas as etapas do desenvolvimento fundamentadas no levantamento dos requisitos aqui apresentados.

## **4. Desenvolvimento da IHM**

Neste capítulo serão apresentadas as ferramentas de software utilizadas no projeto da IHM (IWS) e na implementação do algoritmo de localização das descargas parciais (Matlab). Também será abordado o desenvolvimento da IHM, detalhando a criação das telas, tags utilizadas em todo o projeto, criação de gráficos de tendência e alarmes, uso de animações e script utilizados no tratamento dos dados que serão apresentados ao operador e manipulação do banco de dados.

### **4.1. Ferramentas adotadas: Indusoft Web Studio e Matlab**

Neste projeto fizemos uso de dois softwares, o IWS e o Matlab, onde o IWS foi utilizado no desenvolvimento da IHM, e o Matlab na execução de algoritmos de localização de descargas parciais. O InduSoft Web Studio (ou IWS) é uma poderosa ferramenta que explora as funcionalidades dos sistemas operacionais Microsoft, e permite construir aplicações completas SCADA (Sistemas de Supervisão e Aquisição de Dados) ou IHM (Interface Homem Máquina) para a Indústria de Automação. [7] Além disso, o IWS oferece uma gama de recursos para tornar a aplicação mais robusta e atrativa ao usuário. Dentre esses recursos estão animações, fácil criação de telas, alarmes, gráficos de tendência, scripts, comunicação com outros dispositivos via drivers ou OPC, entre outros recursos.

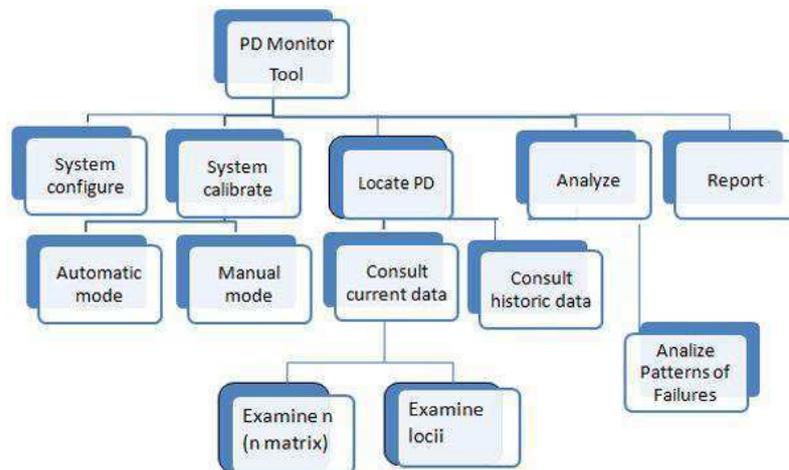
O MATLAB (MATrix LABoratory) trata-se de um software interativo de alta performance voltado para o cálculo numérico. O MATLAB integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente fácil de usar onde problemas e soluções são expressos somente como eles são escritos matematicamente, ao contrário da programação tradicional. [11] O MATLAB é um sistema interativo cujo elemento básico de informação é uma matriz que não requer dimensionamento. Esse sistema permite a resolução de muitos problemas numéricos em apenas uma fração do tempo que se gastaria para escrever um programa semelhante em linguagem Fortran, Basic ou C. Além disso, as soluções dos problemas são expressas quase exatamente como elas são escritas matematicamente.

## 4.2. Desenvolvimento das Telas

A função mais básica do IWS é proporcionar uma tela para o processo. O IWS permite criar telas com dimensões e posições complementares e associá-las a um grupo de telas, para que as mesmas possam ser abertas simultaneamente. Dessa maneira, evita-se a necessidade de criar repetidamente interfaces que devem ser mostradas todo o tempo, criando-se telas de cabeçalho e rodapé. [7]

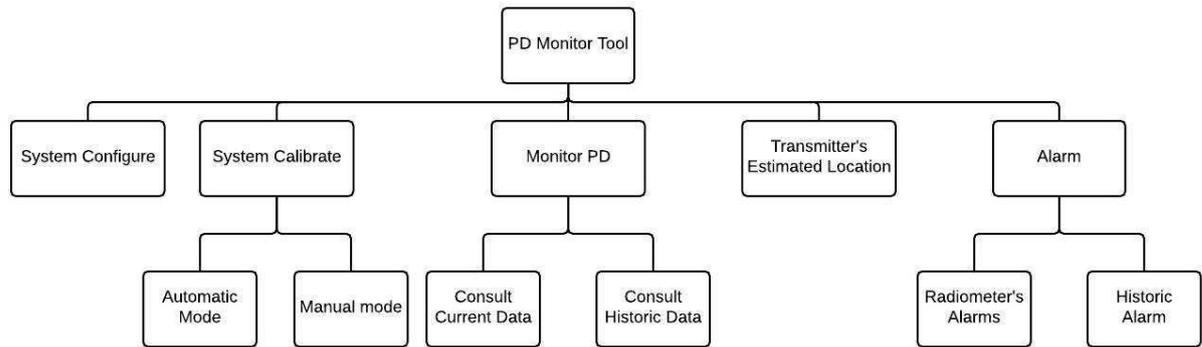
Em um sistema automatizado com IHM, as telas podem representar estados, pequenas partes ou uma visão geral do sistema. Então, é necessário que o desenvolvedor conheça todo o processo que será representado a partir das telas, e tenha um modelo da tarefa que será desempenhada pela IHM bem definido antes de começar a desenvolvê-las.

A proposta do conceito da IHM já havia sido elaborada antes deste projeto e apresentada à equipe do Reino Unido. Neste projeto, poucas alterações foram introduzidas no modelo da tarefa apresentado na figura 11.

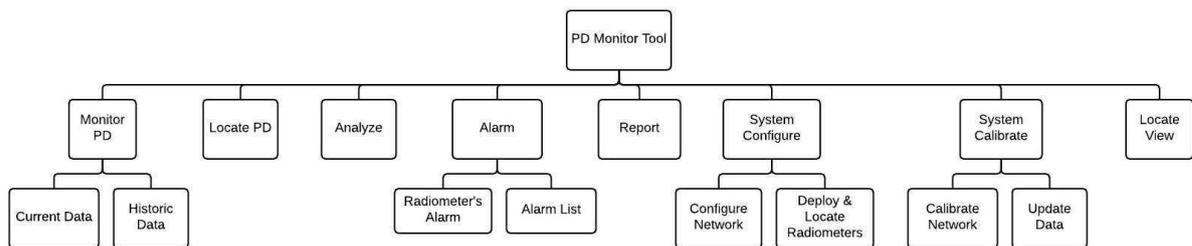


**Figura 11:** Modelo da Tarefa (High Level)

As modificações consistiram em acrescentar algumas funções que resultaram em telas para detalhar informações, principalmente na tarefa de configuração do sistema. Após o refinamento do modelo, foi elaborada a nova versão do modelo da tarefa, apresentada na figura 12 e a hierarquia de telas resultante, ilustrada na figura 13.



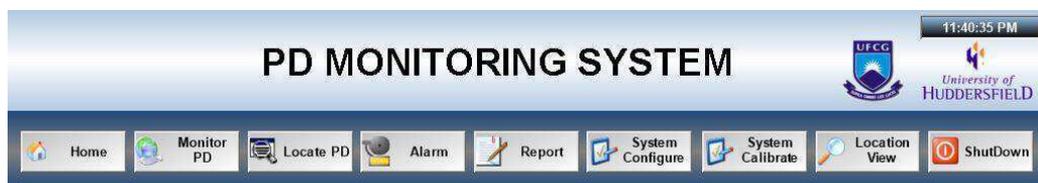
**Figura 12:** Modelo de Tarefa Atualizado



**Figura 13:** Hierarquia de telas

O desenvolvimento das telas da IHM se fundamentou na hierarquia de telas apresentada, na figura 13. O uso do IWS se fundamentou em apostila e vídeos de treinamento [7] disponibilizados pelo fabricante do IWS.

O menu proposto é do tipo horizontal, e utiliza ícones e textos, como ilustrado na figura 14. Nele, é exibida uma área com o nome da IHM, *PD Monitoring System*, e os logotipos das universidades envolvidas no projeto. Na área imediatamente abaixo, temos os botões de navegação entre telas, que foram desenvolvidos de acordo com a hierarquia de telas da figura 13.

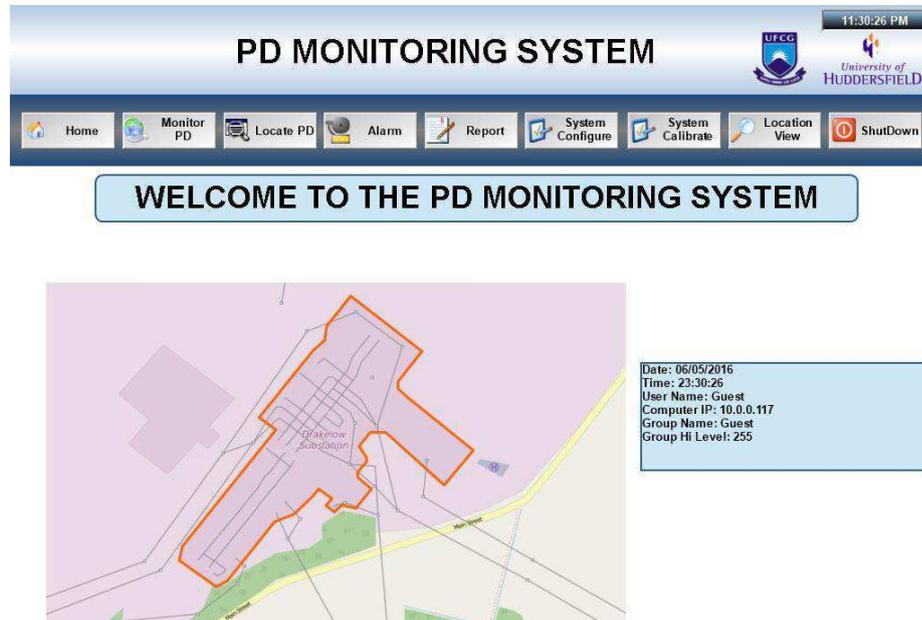


**Figura 14:** Tela com nome da IHM e botões de navegação entre telas

#### Tela de Abertura do Sistema

Em seguida foi desenvolvida a tela de abertura do sistema, mostrada na figura 15 que contém uma imagem da subestação de Drakelow retirada do Google Maps [8], localizada

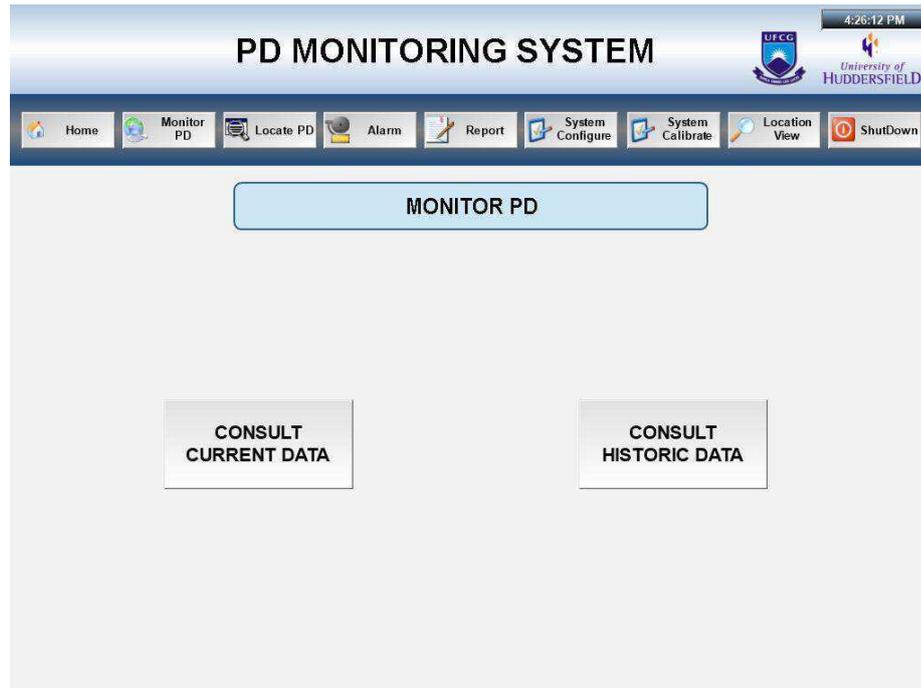
em Derbyshire, Inglaterra, Reino Unido, juntamente com os dados do usuário logado na IHM no momento.



**Figura 15:** Tela inicial

### Tela *Monitor PD*

Dando sequência ao desenvolvimento de telas, foi criada a tela *Monitor PD*, responsável por uma das tarefas mais importantes da IHM, a visualização dos dados coletados por cada radiômetro. Nesta tela, figura 16, temos dois botões que conduzem para as telas de *Current Data* e *Historic Data*, onde os dados atuais referentes a cada radiômetro serão mostrados a partir de uma janela pop-up e os dados históricos serão apresentados para análises de comportamento do sistema.



**Figura 16:** Monitor PD

### *Tela Network Snapshot View*

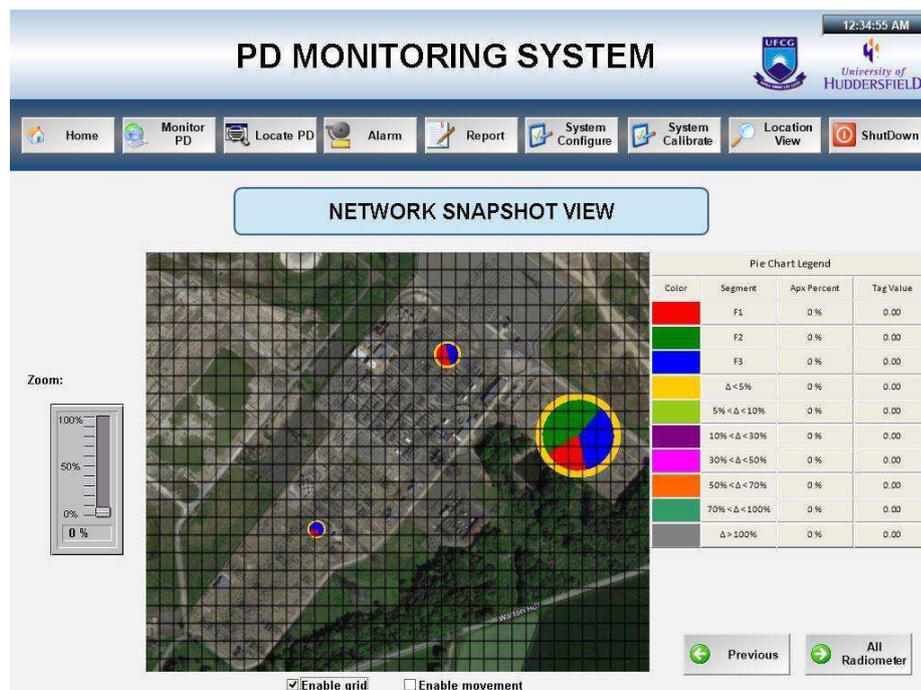
Na tela *Network Snapshot View*, figura 17, temos uma imagem retirada do Google Maps [8] que ilustra a subestação em que os radiômetros serão instalados. Nesta tela, é apresentada a disposição dos radiômetros de acordo com a localização, em coordenadas geográficas, inserida na tela *System Configure*, que será apresentada mais adiante.

A representação utilizada até o momento, e que foi discutida entre as universidades parceiras, é a representação por irradiação de energia, onde o raio da circunferência relativa a cada radiômetro e a parcela de potência relativa a cada uma das fases variam como já foi discutido na seção 2.2 (Representação de Dados Complexos na Engenharia). Sabendo que os dados coletados por cada radiômetro são enviados a IHM em forma de arquivos, e que sua atualização é periódica, de acordo com o *Tu (Time Measurement)* ajustado pelo operador, o valor da potência máxima, para que seja atribuída a essa potência o raio máximo de cada radiômetro, está sempre mudando.

Sendo assim, um quadro informando a porcentagem de crescimento da potência recalculada foi construído, para que o operador tivesse conhecimento do crescimento da nova potência calculada, pois como o raio máximo é fixo e igual a 40 pixels e a potência que é calculada está sempre em crescimento, então identificou-se a necessidade de informar ao

operador, em termos de porcentagem, o quanto a potência medida por cada radiômetro aumentou. Como pode ser visto no quadro da figura 17, as faixas de crescimentos foram divididas por cores, por exemplo, a faixa de 10% a 30% é representada pela cor roxa.

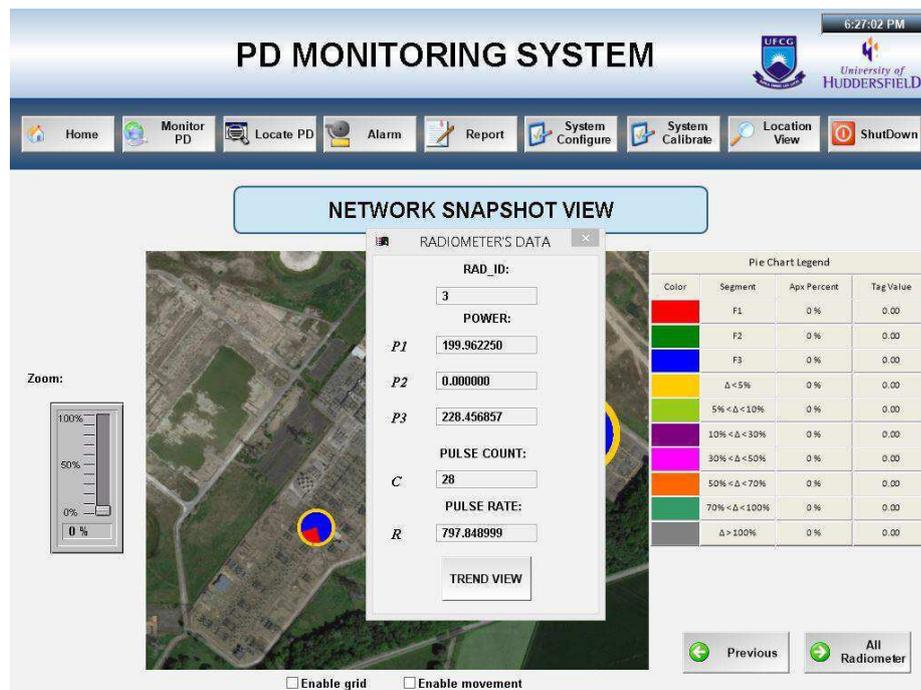
Ainda na tela *Network Snapshot View*, temos as opções de Zoom, habilitar grade, habilitar movimento, botão de voltar para tela anterior e botão para visualizar todos os dados dos radiômetros em uma única tabela. O *slider* de zoom é um dos objetos padrão do IWS que foi utilizado para realizar o zoom in e zoom out. A opção de habilitar a grade faz com que, quando selecionado, uma grade apareça para facilitar a localização dos radiômetros entre os equipamentos da subestação. Já a opção de habilitar movimento possibilita ao usuário movimentar a imagem após realizar um zoom in, por exemplo. Todos os procedimentos de criação dessas opções serão descritos detalhadamente na seção 4.5 (Animações).



**Figura 17:** Dados atuais e disposição dos radiômetros (Current Data)

Ao clicar em um dos radiômetros uma janela pop-up aparece, figura 18, esta tela possui seis campos com dados referentes ao radiômetro que foi solicitado. O primeiro campo, chamado RAD\_ID, contém a identificação do radiômetro que foi clicado. Os campos *P1*, *P2* e *P3*, referem-se a potência medida por este radiômetro para cada uma das fases do sistema. O campo *C*, *Pulse Count*, refere-se a quantidade de pulsos gerados por descargas parciais que foram medidos.

E por fim, o campo *Pulse Rate* diz respeito a razão entre o número de pulso medidos em um período de tempo, onde este período de tempo é ajustado na tela *System Configure*, e corresponde ao intervalo de tempo determinado pelo usuário para realizar o tratamento dos dados de cada radiômetro. O detalhamento destes parâmetros foi feito na seção 2.4 (Descrição dos Dados do Projeto).



**Figura 18:** Janela Pop-up com dados dos radiômetros

O botão *Trend View* direciona o usuário à tela da figura 19, onde são apresentados os dados históricos referentes a apenas o radiômetro da tela pop-up. Os dados históricos que podem ser acessados são os dados da janela pop-up *Radiometer's Data*. Nesta tela o usuário pode escolher o período em que deseja visualizar os dados (Dia, Mês, Ano, Hora, Minuto e Segundo), quais parâmetros deseja visualizar (P1, P2, P3, Pulse Count ou Pulse Rate), ou pode voltar para a tela anterior (*Network Snapshot View*).

Se o usuário optar por visualizar os dados referentes a todos os radiômetros em um determinado período de tempo, ele deve clicar sobre o botão *All Radiometers*, que o direcionará para tela da figura 20. Nesta tela temos uma tabela contendo todos os parâmetros medidos por cada radiômetro. Tais parâmetros já foram detalhados na seção 2.4 (Descrição dos Dados do Projeto), sendo assim não há necessidade de realizar grandes comentário sobre eles nesta seção.

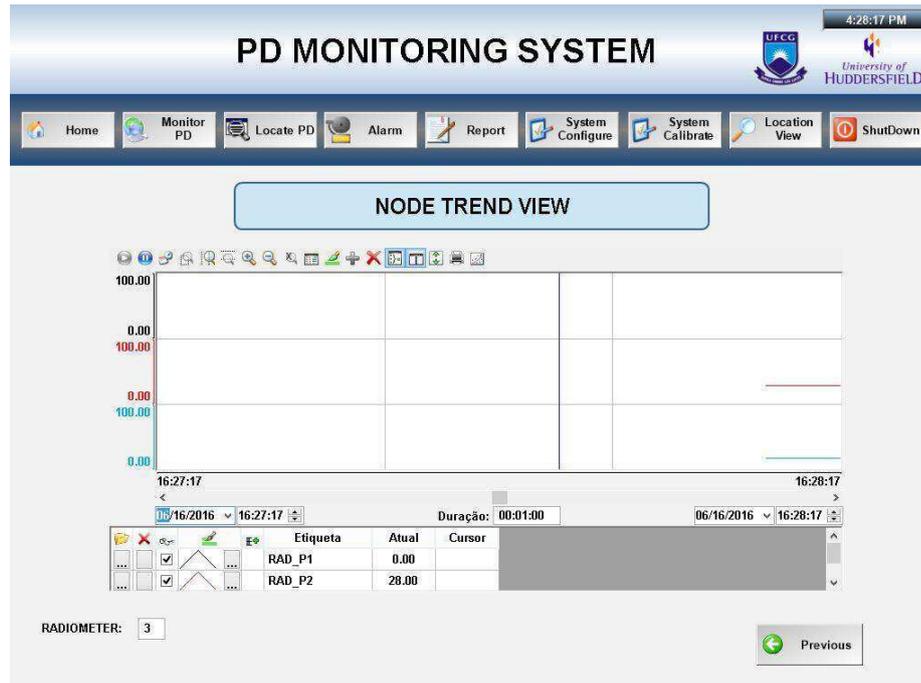


Figura 19: Dados históricos de um radiômetro

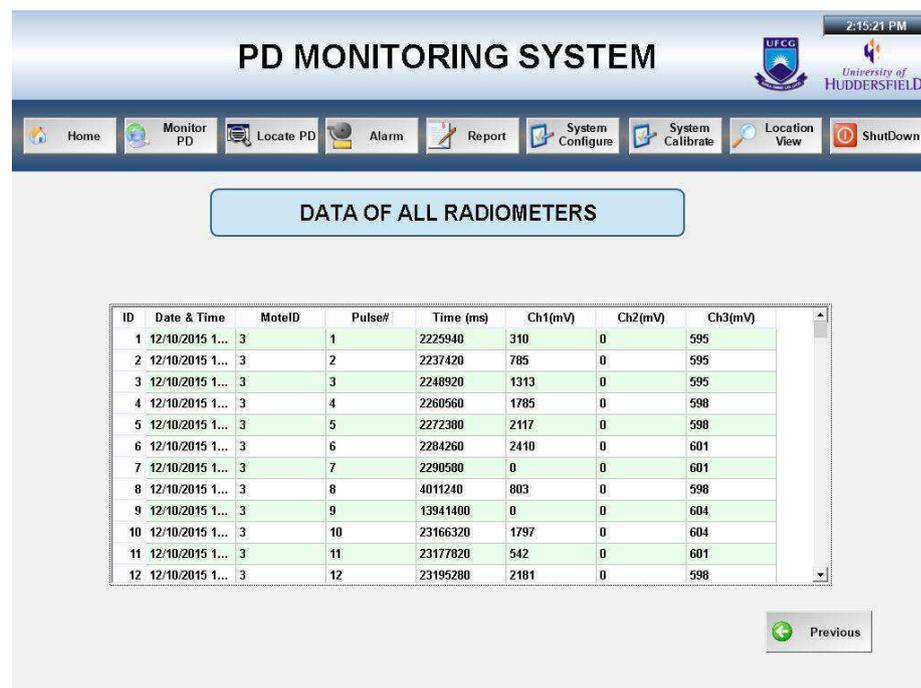
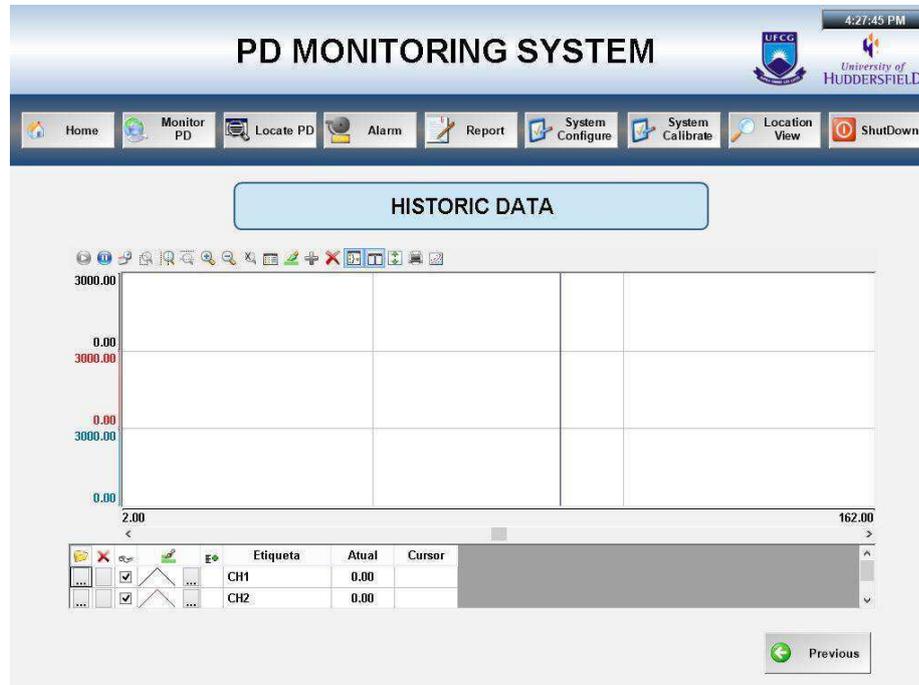


Figura 20: Tabela contendo todos os dados dos radiômetros

### Tela *Historic Data*

Na tela *Historic Data*, figura 21, são apresentados os dados históricos de potência medida por cada um dos radiômetros em cada uma das fases, em forma de gráficos de tendência. Nesta tela o usuário pode escolher se deseja visualizar os dados históricos relativos a um ou a todos os radiômetros, além de poder visualizar uma ou todas as fases que estão sendo

monitoradas por cada radiômetro. Esta escolha pode ser feita a partir da seleção dos parâmetros desejados no quadro mostrado abaixo dos gráficos de tendência. A criação de gráficos de tendência e como os dados são tratados para serão representados graficamente será discutida mais detalhadamente na seção 4.6 (Gráficos).



**Figura 21:** Dados históricos dos radiômetros

### Tela *Configure Network*

Após a criação das telas descritas anteriormente, iniciamos a criação da tela *Configure Network*, figura 22, onde é possível realizar a configuração do sistema no qual os radiômetros estão inseridos. Nesta tela temos o nome da subestação (Substation Name), a região na qual está localizada (Location), a localização geográfica (Latitude e Longitude), as coordenadas geográficas da instalação que determinará a origem do sistema de visualização dos radiômetros na tela *Location View* (Installation (x,y) Coordinate System Origin (0,0)), o ângulo de referência para o sistema de coordenadas, e o número de radiômetros existentes na subestação.

**Figura 22:** Tela de configuração do sistema

Clicando no botão *Next*, o usuário é direcionado a tela da figura 23, onde ele poderá inserir a localização geográfica de cada radiômetro existente no sistema. Dentre os parâmetros da tabela, temos a latitude e longitude com campos separados em graus, minutos e segundos. Ainda como parâmetros da tabela, temos a localização em relação ao eixo x, y em metros, onde a referência é a localização geográfica fornecida pelo usuário no campo *Installation (x,y) Coordinate System Origin (0,0)* da tela *Configure Network*. Após o preenchimento da localização dos radiômetros, esta localização será informada na tela pop-up *Radiometer's Data*, no campo *Longitude* e *Latitude*.

Por fim, nessa mesma tela, o operador poderá definir o valor do limite superior da potência medida por cada radiômetro para que alarmes sejam acionados caso a potência ultrapasse o valor estipulado. Foi definido que o usuário poderia ajustar estes valores, visto que a potência gerada por descargas parciais que um equipamento suporta pode ser diferente da potência suportada por outro equipamento.

The screenshot displays the 'PD MONITORING SYSTEM' interface. At the top right, the time is 6:30:19 PM and the University of Huddersfield logo is visible. The navigation bar contains icons for Home, Monitor PD, Locate PD, Alarm, Report, System Configure, System Calibrate, Location View, and ShutDown. The main section is titled 'DEPLOY & LOCATE RADIOMETERS' and contains a table with the following data:

ID	RADIOMETE...	LATITUDE (°)	LATITUDE (°)	LATITUDE (°)	LONGITUDE ...	LONGITUDE (°)	LONGITUDE ...	X (m)	Y (m)	Pmax (ALA...
1	3	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0

At the bottom right of the main content area, there are two buttons: 'Previous' and 'Location View'.

**Figura 23:** Tela para inserção da localização dos radiômetros

Clicando no botão *System Calibrate*, o usuário será direcionado a tela *Calibrate Network*, figura 24, onde são realizados os ajustes referentes a calibração que deve ser feita no sistema de medições. Esta calibração é de grande importância para que seja feita uma triangulação para determinar a localização dos radiômetros e das fontes de descargas parciais. Esta triangulação é feita emitindo um sinal por um dos radiômetros, que é configurado como fonte, e a partir da emissão desse sinal os demais radiômetros podem ser localizados. Então o usuário deve fornecer o tipo de comunicação existente no sistema (*Network*), a localização da fonte de emissão do sinal, a intensidade da fonte, e determinar se o sistema deve ou não ser calibrado selecionando uma das opções no campo *Calibrate* e clicar sobre o botão *Update All*.

The screenshot shows the 'PD MONITORING SYSTEM' interface. At the top right, the time is 4:26:44 PM and the University of Huddersfield logo is visible. The navigation menu includes: Home, Monitor PD, Locate PD, Alarm, Report, System Configure, System Calibrate, Location View, and ShutDown. The main content area is titled 'CALIBRATE NETWORK' and contains the following elements:

- A 'CALIBRATE NETWORK' button at the top.
- A 'NETWORK' dropdown menu.
- A 'LAST CALIBRATION' text input field.
- A 'SOURCE LOCATION' text input field.
- A 'SOURCE INTENSITY' text input field.
- A 'CALIBRATE' button with radio buttons for 'YES' and 'NO'.
- A 'Next' button with a green arrow at the bottom right.

**Figura 24:** Tela para calibração do sistema

#### Tela *Update Data*

A tela *Update Data*, figura 25, faz parte da calibração do sistema, onde o usuário é direcionado para ela clicando no botão *Next* da tela *Calibrate Network*. Esta tela possui dois campos de extrema importância para o tratamento e atualização dos dados fornecidos a IHM pelo sistema de captação de dados dos radiômetros. O campo *Automatic Update Period, Tu*, diz respeito ao período de tempo em que o pacote de dados referente a cada radiômetro deve ser atualizado, visto que os dados recebidos pela IHM são em forma de arquivos .xls (Microsoft Excel), e que a faixa de valores aceitáveis é de 60 a 86400 segundos. O valor padrão para esse campo é de 3600 segundos.

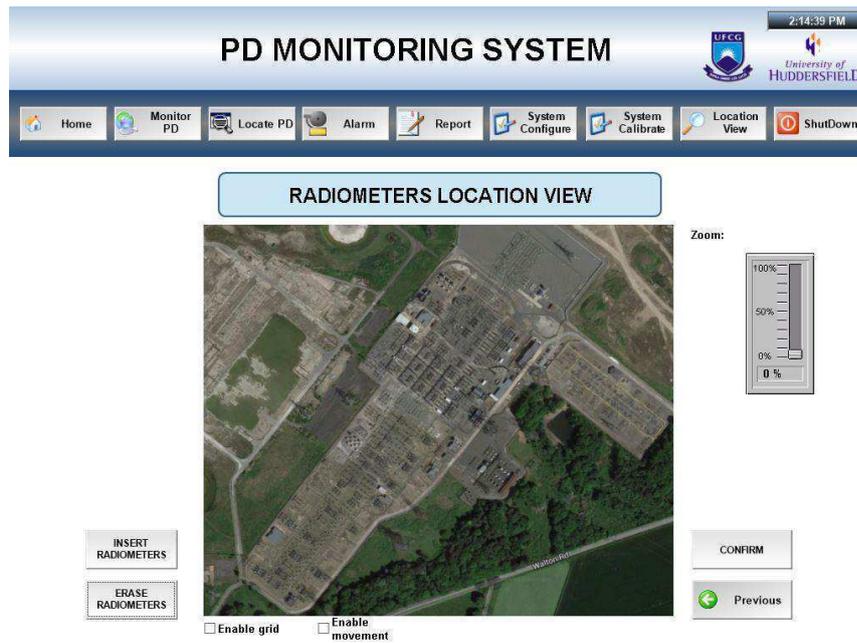
No campo *Measurement Duration, Tm*, é possível definir o período de tempo em que os dados serão analisados, ou seja, o usuário define um valor entre 0,02 segundos e o tempo definido no campo *Automatic Update Period*, e este tempo definido será utilizado no tratamento dos dados, como mostrado na seção 2.4 (Descrição dos Dados do Projeto). O valor padrão para este campo é de 1 segundo. Ainda nesta tela, o usuário pode parar ou iniciar as medições (*Start Measurement* e *Stop Measurement*) que estão sendo realizadas por cada radiômetro, além de poder voltar para tela anterior ou acessar a tela *Network Snapshot View* que contém os radiômetros existentes na subestação nos botões *Back* e *Current Data*, respectivamente.

A caixa de seleção *Automatic Update* bloqueia a inserção de novos parâmetros para que sejam utilizados os valores padrão de cada variável. Já a caixa de seleção *Manual Update* permite que o usuário insira novos valores para os parâmetros acima detalhados.

**Figura 25:** Tela Update Data para calibração do sistema

### Tela *Location View*

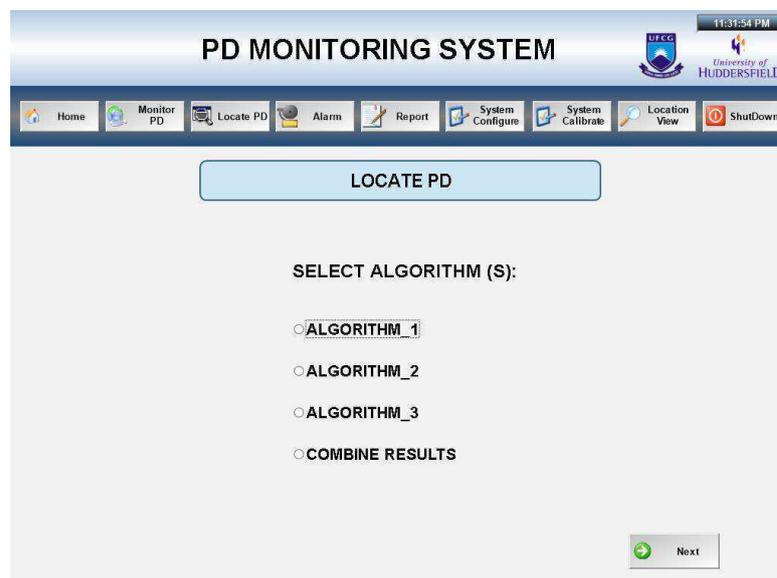
Dando continuidade ao desenvolvimento de telas, foi criada a tela *Location View*, figura 26, onde é possível inserir ou remover os radiômetros a partir dos botões *Insert Radiometers* e *Erase Radiometers*, respectivamente. Esta tela tem muita semelhança com a tela *Network Snapshot View*, mas o diferencial é que nesta tela não é possível visualizar os dados referentes a cada radiômetro. Clicando no botão *Insert Radiometers*, o usuário é direcionado a tela *Deploy & Locate Radiometers* que faz parte do grupo de telas de configuração do sistema, onde ele inserirá as coordenadas geográficas para cada radiômetro. Clicando no botão *Erase Radiometers*, o usuário pode escolher quais radiômetros removerá, ou pode escolher a remoção completa.



**Figura 26:** Tela Location View para inserção e remoção dos radiômetros

### Tela *Locate PD*

Nesta tela o usuário pode selecionar o algoritmo para localização das descargas parciais, ou realizar uma combinação deles selecionando a opção *CombineResults*. Ao selecionar uma das opções e prosseguir, os dados necessários são coletados pelo Matlab para processamento e localização estimada das descargas parciais. Vale ressaltar que os algoritmos ainda estão em processo de desenvolvimento pela equipe do UK. A tela para representar graficamente a localização das DP será projetada em uma etapa futura.



**Figura 27:** Tela de escolha do algoritmo para localização das DPs

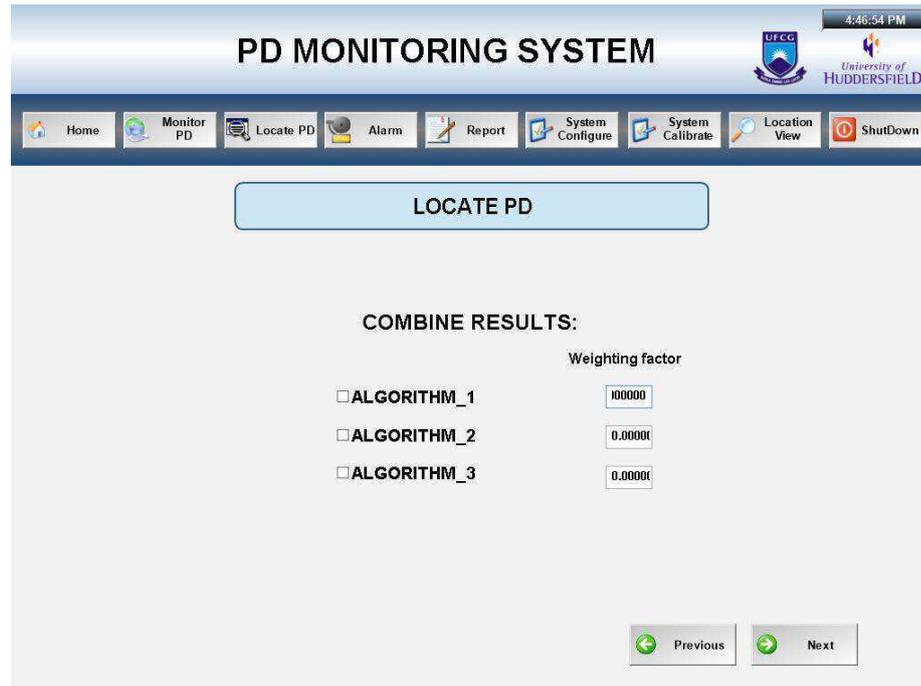


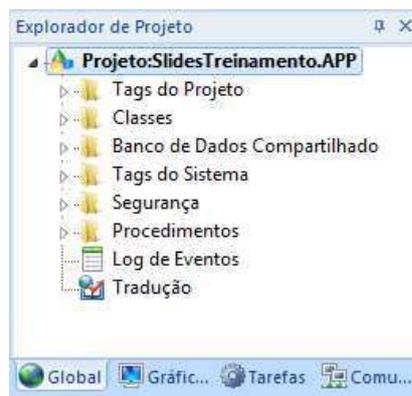
Figura 28: Tela de combinação dos algoritmos

### 4.3. Conexão da IHM - Definição dos Tags no IWS

Tags constituem o núcleo central de qualquer projeto do IWS. Tags são variáveis usadas pelo IWS para receber e armazenar os dados obtidos a partir da comunicação com dispositivos no chão de fábrica, a partir dos resultados dos cálculos e funções, e de entrada do usuário. Por sua vez, as tags podem ser utilizadas para mostrar informações nas telas, para manipular objetos da tela, e para controlar tarefas em execução.[7] As Tags podem ser utilizadas como variáveis internas ou externas, pelo ponto de vista da IHM. As variáveis internas são utilizadas apenas pela IHM e as externas são variáveis já existentes no CLP ao qual a IHM está conectada.

As variáveis internas devem ser utilizadas para auxiliar na definição da apresentação das telas para o operador, ou seja, são TAGs que controlam elementos dentro da própria IHM. Já as variáveis externas devem ser utilizadas para receber dados do CLP ou enviar dados para o CLP. Essas TAGs serão utilizadas para demonstrar para o operador o estado em que o sistema se encontra, pela tela da IHM, enviar alertas e alarmes para o operador ou para ativar transições, alterando o estado do sistema. Portanto, saber manipular as variáveis externas de maneira correta é de extrema importância, visto que a alteração incorreta pode ser prejudicial ao sistema.

As TAGs são mais do que simples variáveis. O IWS inclui um gerenciador de banco de dados em tempo real que fornece um número de funções sofisticadas, como a impressão da hora de qualquer alteração de valor, verificando valores de TAG contra os valores mínimo e máximo de tempo de execução, comparando os valores de TAGs para limites de alarmes, e assim por diante. Uma TAG do IWS tem valor e várias propriedades que podem ser acessados, alguns em desenvolvimento, enquanto outros só na execução. [7]As TAGs são organizadas na aba global do explorador de projeto, como mostra a figura 29.



**Figura 29:** Categorias de TAGs

Dentre as categorias de TAGs existentes no IWS, temos as tags do projeto que são criadas durante o desenvolvimento do projeto e que geralmente são usadas em telas, ler e escrever um dado, de controle, e auxiliares para executar cálculos sobre tags ou banco de dados do projeto. As TAGs de banco de dados compartilhado são criadas em um programa externo e depois importadas para o IWS, mas não podem ser editadas pelo IWS, apenas pelo programa original. As TAGs do sistema são predefinidas como funções predeterminadas que são usadas para tarefas do supervisor IWS.

Em relação a sintaxe das TAGs, elas podem ser criadas com letras, números e o caractere de sublinhado, devem ser únicas, e tem um tamanho máximo de 255 caracteres. Os nomes das tags devem ser diferentes dos nomes das tags do sistema e funções matemáticas. Os tipos de dados que as tags podem assumir no IWS são:

- **Booleana:** Com valores possíveis de “0” (falso) ou “1” (verdadeiro). Geralmente utilizadas para ligar e desligar objetos ou para abrir e fechar objetos.
- **Inteira:** Número inteiro de 32-bit sinalizado. Normalmente utilizado para contagem de números inteiros ou ajustar valores de números inteiros.

- **Real:** Número real de 64-bit sinalizado. Utilizado para medições ou valores decimais/fracionários.
- **String:** Com até 1024 caracteres, contendo letras, números ou caracteres especiais. Suporta caracteres ASCII e UNICODE.

Outra categoria de tags é a Tag vetor, que consiste em um conjunto de tags com o mesmo nome e mesmo tipo, onde as tags são organizadas na forma de uma matriz n linhas por 1 coluna. O tamanho máximo para o vetor é de 16.384, sendo necessário que o desenvolvedor defina o tamanho do vetor. Cada posição do vetor, a começar da posição 0, é uma tag do projeto. É importante utilizar ao máximo a tag vetor, pois otimiza a memória e minimiza o tempo de configuração das variáveis. Também é possível determinar o tamanho do vetor como n, logo o vetor pode variar de tamanho de acordo com a necessidade.

Caso haja variáveis no projeto que possuam múltiplos atributos de diferentes tipos, o desenvolvedor pode utilizar a categoria de Tag classe que é um modelo composto por tags de diferentes tipos. Desta forma é possível encapsular os atributos referentes a um item do projeto, otimizando o uso da memória e mantendo a organização do projeto. Vale ressaltar que os nomes das classes devem ser únicos, o desenvolvedor precisa definir os membros e os seus tipos, onde o número máximo de membro é de 4096. O membro de uma classe não pode pertencer à outra classe, mas o desenvolvedor pode criar membros de mesmo nome em classes diferentes.

Por fim, as Tags indiretas finalizam a categoria de tipos de tags existentes no IWS. Essas tags são utilizadas para realizar o acesso indireto de tags do banco de dados, ou seja, elas podem fornecer um ponteiro para qualquer outro tipo de tag do projeto. Para visualizar o conteúdo de uma tag indireta deve-se colocar o “@” antes do nome da tag (ex.: @TagIndireta).

Neste projeto foram utilizados vários tipos de tags, como mostra a figura 30, onde optou-se por utilizar tags do tipo vetor e classe para otimizar a memória e manter a localização de uma tag mais simples, dada a organização que as tags vetor e classe fornecem. Na figura 31 temos a classe RAD com os atributos referentes aos radiômetros, como já foram descritos na seção 2.4 (descrição dos Dados do Projeto). Já na figura 32, temos a classe Substation com os atributos de localização geográfica (Longitude e Latitude), o ângulo de referência de rotação do sistema e o número de radiômetros existentes na subestação.

	Nome	Vetor	Tipo	Descrição	Escopo
7	Radiometers_Data	9	Inteira	Dados para teste ...	Local
8	Algorithm_Result	4	Real	Resultado do proc...	Local
9	Combine_Results	2	Inteira	Algoritmos que for...	Local
10	Algorithm_Chosen	0	String	Algoritmo(s) escol...	Local
11	AutomaticManual...	0	Booleana	Automatic = 1, Man...	Local
12	sEtiqueta_Update...	0	String	Etiqueta da caixa d...	Local
13	nPosicao_Update...	0	Inteira	Posição dos elem...	Local
14	RAD_DATA	0	RAD	Dados referentes ...	Local
15	UPDATE_BD	0	Booleana	Variável para identi...	Local
16	Salvar_dados	0	Booleana	Salvar ultimo regis...	Local
17	SUBSTATION	0	Substation		Local
18	Red_Symbol	2	Booleana	Representa uma e...	Local
19	Green_Symbol	2	Booleana	Representa uma b...	Local
20	Tu	0	Real	Automatic Update ...	Local
21	Tm	0	Real	Measurement Dur...	Local

Figura 30: Tags do projeto

	Nome	Tipo	Descrição
1	CH1	Real	
2	CH2	Real	
3	CH3	Real	
4	PulseCount	Inteira	
5	PulseRate	Real	
6	LONG_graus	Real	
7	LONG_min	Real	
8	LONG_seg	Real	
9	LAT_graus	Real	
10	LAT_min	Real	
11	LAT_seg	Real	
12	ID	Inteira	
13	X	Real	
14	Y	Real	
15		Inteira	

Figura 31: Classe RAD (Atributos dos radiômetros)

	Nome	Tipo	Descrição
1	Name	String	
2	LAT_GRAUS	Real	
3	LAT_MIN	Real	
4	LAT_SEG	Real	
5	LONG_GRAUS	Real	
6	LONG_MIN	Real	
7	LONG_SEG	Real	
8	SYSTEM_ROTATION	Real	
9	NUMBER_RADIOMET...	Inteira	
*		Inteira	

Figura 32: Classe Substation (Atributos da subestação)

#### 4.4. Uso de Animações na IHM

Nesta seção descreveremos as animações que foram utilizadas para que o projeto se tornasse mais dinâmico, fazendo com que a experiência do operador seja a melhor possível. As animações podem ser utilizadas em um objeto de telas ou em um grupo de objetos, e permitem que as propriedades do objeto sejam modificadas em tempo real de acordo com os valores das tags. [7] O IWS possui 8 tipos de animações, são elas:

- **Comando:** Permite que o usuário clique no objeto ou pressione uma tecla pré-definida para executar um comando definido pelo desenvolvedor em tempo de execução. [7]
- **Hiperlink:** Permite que o operador clique sobre um objeto durante a execução para abrir a URL configurada no navegador padrão. [7]
- **Gráfico de barras:** Permite a configuração de um objeto como gráfico de barras. [7]
- **Texto com link de dados:** Permite inserir e exibir valores de tags em tempo real ou entrar com valores via teclado em objetos de texto. [7]
- **Cor:** Modificação de cor de um objeto durante a execução de acordo com o valor de uma tag ou expressão. [7]
- **Visibilidade/Posição:** Permite mover um objeto na horizontal e/ou vertical, mostrar ou ocultar objeto, durante tempo de execução. [7]
- **Redimensionar:** Permite aumentar ou diminuir o tamanho de um objeto durante a execução. [7]
- **Rotação:** Permite rotacionar objeto na tela, a partir de diferentes pontos de referência, durante execução. [7]

Neste projeto foram utilizadas as animações de comando, visibilidade/posição e redimensionar. O uso destas animações foi em sua maioria na tela Network Snapshot View, figura 17, que contém os objetos que representarão os dados fornecidos por cada radiômetro, e é a tela responsável por auxiliar o operador na tomada de decisões em relação ao nível das descargas parciais que estão sendo captadas por cada radiômetro.

Inicialmente para estabelecer a navegação entre as telas da IHM foi utilizada a animação de comando, onde foram inseridos os botões referentes a cada tela, como mostra a figura 14, e em seguida a animação comando foi selecionada na barra gráficos. A partir desta opção de animação o desenvolvedor pode optar em utilizar diversos comandos, como abrir ou fechar telas, executar um script desenvolvido em VBscript, executar um comando built-in

(Linguagem interna do IWS), entre outros comandos possíveis. A animação de comando foi utilizada para diversas aplicações neste projeto, como abertura e fechamento de telas, execução de scripts, ajuste de tags, etc.

Além da animação de comando, foram utilizadas as animações de visibilidade/posição e redimensionar nas telas das figuras 17 e 26. Nestas telas a animação de visibilidade/posição foi utilizada para que o operador pudesse habilitar ou desabilitar a grade, e a animação de redimensionar foi utilizada para que o operador pudesse mover e aplicar o zoom a figura. Desta forma, com a ajuda de ferramentas de simples implementação foi possível dinamizar as telas da IHM, fornecendo ao operador uma experiência de utilização mais satisfatória.

#### **4.5. Definição de Alarmes**

O objetivo principal do Alarme é informar o operador de qualquer problema ou situação anormal durante o processo para que ele possa tomar uma ação corretiva. É executado em Background, e gera os alarmes, porém não mostra os mesmos (necessário utilizar o Objeto de Alarme/Evento). Permite configurar grupos de alarmes e definir as mensagens de alarmes geradas pelo projeto e pode gerar arquivos de histórico. [7]

Possui 2 tipos de Alarmes principais:

- Alto (HiHi, Hi): Gera o alarme se o valor for igual ou superior ao limite especificado;
- Baixo (LoLo, Lo): Gera o alarme se o valor for igual ou inferior ao limite especificado;

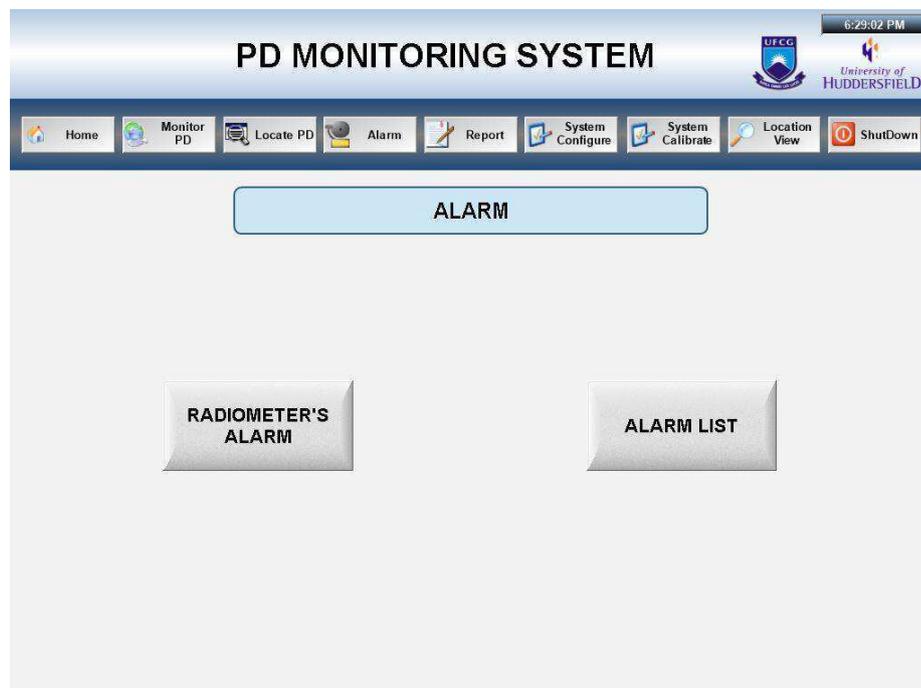
Possui 3 tipos de Estado:

- Ativos: significa que a condição de alarme é verdadeira e que ainda não foi reconhecido ou normalizado;
- Reconhecidos: significa que o alarme ainda está ativo, e foi reconhecido por algum operador;
- Normalizado: significa que a condição de alarme não é mais verdadeira, porém o alarme ainda não foi reconhecido;

Neste projeto foram desenvolvidas duas telas de alarmes, uma para que o operador possa verificar que tipos de alarmes estão ativos, podendo reconhecê-los, e outra para o histórico de todos os alarmes que ocorreram desde o início da execução da IHM. O único tipo

de alarme necessário para este projeto será o de alto, onde o operador poderá definir o limite superior da potência medida por cada radiômetro a partir da tela de configuração mostrada na figura 23. Foi definido que o operador ajustaria o valor máximo da potência medida por cada radiômetro, visto que este valor pode mudar dependendo da localização do radiômetro, pois ele pode estar próximo de um equipamento que possua um limite máximo de potência diferente de outros.

Na figura 31 temos a tela principal dos alarmes, onde o operador pode escolher se deseja visualizar os alarmes recentemente ocorridos, ou se deseja visualizar o histórico dos alarmes ocorridos no sistema. Na figura 32 temos a tela dos alarmes que ocorreram recentemente, onde o usuário pode reconhecê-los e/ou normalizá-los. O único alarme definido para teste foi o limite superior da potência medida por cada radiômetro, onde o operador deve inserir este valor na tela de configurações do sistema, figura 23, e caso a potência ultrapasse esse valor, um alarme é disparado. E na figura 33 temos a tela contendo o histórico dos alarmes ocorridos.



**Figura 33:** Tela de alarmes

**PD MONITORING SYSTEM**

6:29:24 PM  
UFCG  
University of HUDDERSFIELD

**RADIOMETER'S ALARM**

Hora de ativ...	Nome da Tag	Mensagem
06/02/2016 18:26:11	RAD_DATA[...]	Very high power - CH2 (RAD0)
06/02/2016 18:26:11	RAD_DATA[...]	High power - CH2 (RAD0)
06/02/2016 18:26:11	RAD_DATA[...]	High power - CH3 (RAD0)
06/02/2016 18:26:11	RAD_DATA[...]	Very high power - CH3 (RAD3)
06/02/2016 18:26:11	RAD_DATA[...]	High power - CH3 (RAD3)
06/02/2016 18:26:11	RAD_DATA[...]	High power - CH1 (RAD3)

Previous

Figura 34: Tela de alarmes recentes

**PD MONITORING SYSTEM**

6:29:47 PM  
UFCG  
University of HUDDERSFIELD

**ALARM LIST**

Hora de ativ...	Nome da Tag	Mensagem
06/02/2016 18:26:11	RAD_DATA[...]	High power - CH1 (RAD3)
06/02/2016 11:49:29	RAD_DATA[...]	High power - CH3 (RAD3)
06/02/2016 11:49:29	RAD_DATA[...]	High power - CH1 (RAD3)
06/02/2016 11:49:29	RAD_DATA[...]	Very high power - CH3 (RAD3)
06/02/2016 11:49:29	RAD_DATA[...]	High power - CH3 (RAD0)
06/02/2016 11:49:29	RAD_DATA[...]	Very high power - CH2 (RAD0)
06/02/2016 11:49:29	RAD_DATA[...]	High power - CH2 (RAD0)
06/02/2016 11:30:20	RAD_DATA[...]	Very high power - CH2 (RAD0)
06/02/2016 11:30:20	RAD_DATA[...]	High power - CH2 (RAD0)
06/02/2016 11:30:20	RAD_DATA[...]	High power - CH3 (RAD0)
06/02/2016 11:30:20	RAD_DATA[...]	Very high power - CH2 (RAD0)
06/02/2016 11:30:20	RAD_DATA[...]	High power - CH3 (RAD0)
06/02/2016 11:30:20	RAD_DATA[...]	High power - CH2 (RAD0)
06/02/2016 11:30:20	RAD_DATA[...]	High power - CH3 (RAD3)
06/02/2016 11:30:20	RAD_DATA[...]	Very high power - CH3 (RAD3)

Previous Radiometer's Alarms

Figura 35: Tela de histórico de alarmes

## 4.6. Desenvolvimento de Scripts

Os scripts contidos neste projeto foram desenvolvidos em VBScript (acrônimo de Microsoft Visual Basic Scripting Edition), linguagem padrão utilizada pelo IWS. Estes scripts

foram utilizados para implementar os algoritmos de tratamento de dados da IHM, ou seja, realizar o cálculo dos parâmetros apresentados na seção 2.4 deste trabalho a partir dos dados coletados (raw data) por cada radiômetro, armazenados no BD.

Outra linguagem utilizada neste trabalho foi a linguagem de manipulação de dados SQL (Structured Query Language, ou Linguagem de Consulta Estruturada). A linguagem SQL é um grande padrão de banco de dados. Isto decorre da sua simplicidade e facilidade de uso. Ela se diferencia de outras linguagens de consulta a banco de dados no sentido em que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele. Ela é uma linguagem declarativa em oposição a outras linguagens procedurais. Isto reduz o ciclo de aprendizado daqueles que se iniciam na linguagem.

Sendo assim, a linguagem SQL foi utilizada para consulta e alteração de dados no BD, criação de tabelas, inserção de dados nas tabelas, suporte na determinação dos parâmetros dos radiômetros. Os scripts utilizados estão disponíveis no anexo I. Segue uma breve descrição sobre os scripts desenvolvidos:

- **CRIAR E INSERIR DADOS EM UMA TABELA**

Neste script utilizamos a linguagem de manipulação de dados para criar novas tabelas dentro do BD e separar os dados de cada radiômetro, facilitando assim sua manipulação. A função \$DBExecute() permite que qualquer função SQL seja utilizada no IWS.

- **DETERMINAÇÃO DOS PARAMETROS DE CADA RADIOMETRO**

Tratamento dos dados fornecidos por cada radiômetro para exibição ao operador. Implementação das equações apresentadas na seção 2.4 deste trabalho.

- **REPRESENTAÇÃO DOS RADIÔMETROS (Raio e setores das circunferências)**

Neste script foi implementada a estratégia de representação dos dados proposta na seção 2.2 deste trabalho.

#### **4.7. Comunicação Matlab/Simulink e Indusoft**

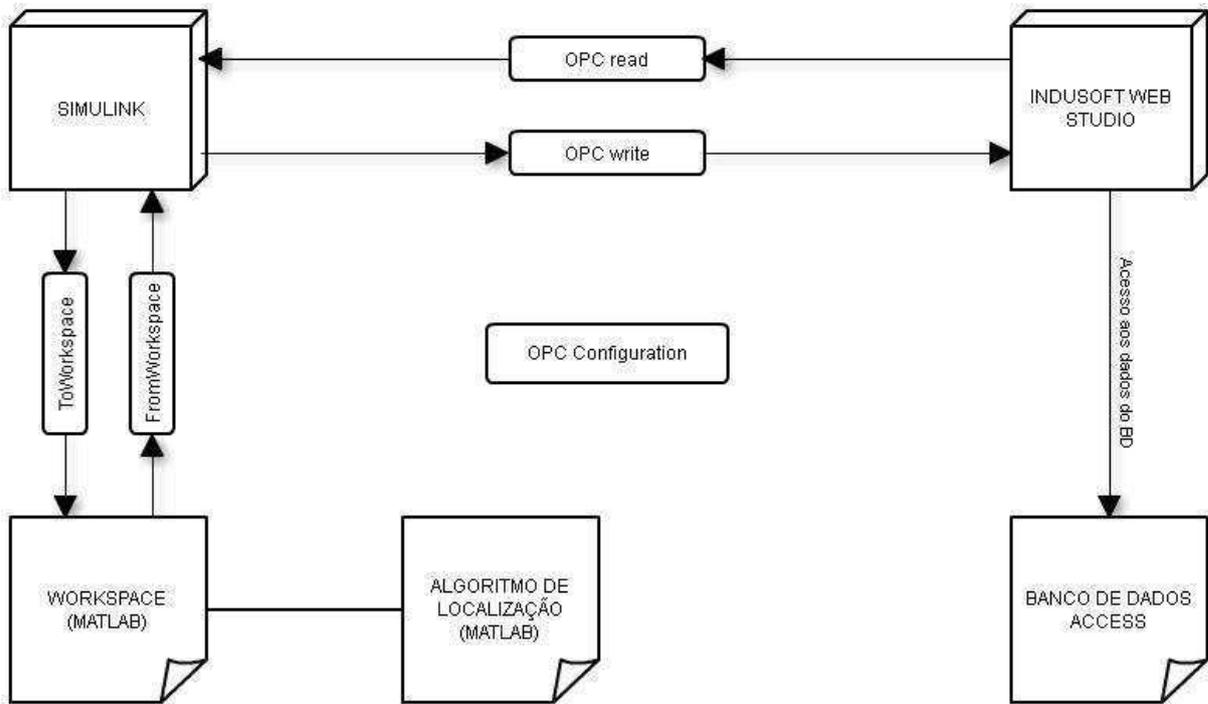
Para possibilitar a integração dos diferentes equipamentos de campo com os sistemas supervisórios da planta, no início do desenvolvimento das redes de automação ficou a cargo de cada um dos fabricantes o desenvolvimento do conjunto de regras, procedimentos e leis que iriam gerir a transferência de dados entre os equipamentos de automação, mais conhecidos como protocolos de comunicação.[9] Mas esse desenvolvimento individual dificultou a integração das plantas industriais, fazendo com que as indústrias tivessem que desenvolver drivers de comunicação por décadas. Até que em 1995 diversas empresas se reuniram para criar um padrão único, baseado na junção de duas tecnologias da Microsoft® já utilizadas pelas indústrias da época, a *OLE (Object Linking and Embedding)* e o *DCOM (Distributed Component Object Model)* [10]. Para este novo protocolo foi atribuído o nome de *OPC (OLE for Process Control)*.

De acordo com Fonseca [10], há duas funcionalidades principais do protocolo: a tecnologia OLE caracteriza-se por integrar diferentes aplicações de automação a uma plataforma Windows, solucionando também os problemas de desempenho e confiabilidade apresentados pelo padrão antecessor, o DDE (Dynamic Data Exchange). Já a tecnologia DCOM permite a implementação de aplicações distribuídas em uma rede industrial através da arquitetura cliente-servidor. Fonseca [10] define esta arquitetura como sendo uma relação dinâmica e completa entre o cliente, que necessita dos dados e pode buscá-los em diversos servidores (armazenagem), e os servidores que tem a liberdade para distribuir os dados coletados a vários clientes.

Neste projeto o protocolo OPC foi utilizado para realizar a comunicação entre o IWS e o Matlab/Simulink. Esta comunicação é de grande importância, tendo em vista que a universidade parceira da UFCG desenvolveu um algoritmo de localização estimada das descargas parciais. Este script foi desenvolvido no ambiente Matlab e necessita que o operador insira a localização dos radiômetros em coordenadas cartesianas (X, Y), onde a origem do sistema é definida na tela de configuração, figura 19, no campo *Instalation Coordinate (x, y) System Origin (0, 0)*.

Então, como essas coordenadas são inseridas em uma das telas da IHM, após sua inserção o Matlab deve ser capaz de acessar essas coordenadas armazenados em Tags do IWS para realizar o cálculo de localização estimada das descargas parciais. Após o a realização deste cálculo, o Matlab precisa salvar as coordenadas das descargas parciais em Tags do IWS para que um gráfico possa ser plotado informando ao operador a provável localização das descargas

parciais que estão ocorrendo com maior intensidade. Um fluxograma que auxilia a compreensão do fluxo de dados entre o IWS e Matlab é mostrado na figura 34.



**Figura 36:** Fluxo de dados entre Matlab/Simulink e IWS

A comunicação pode ser realizada a partir do servidor OPC do IWS, *Studio.Scada.OPC.3*, que torna possível o acesso as Tags do IWS. Todo o processo de configuração da comunicação entre o OPC servidor (IWS) e o OPC cliente (Matlab/Simulink), leitura e escrita de Tags, pode ser realizado a partir do OPC toolbox existente na biblioteca do Matlab/Simulink. Para configurar a comunicação, figura 35, inicialmente deve-se acrescentar o bloco *OPC Configuration* ao projeto do Simulink, e acessando as propriedades desse bloco pode-se adicionar um ou vários servidores clicando na opção *Configure OPC Clients > Add*.

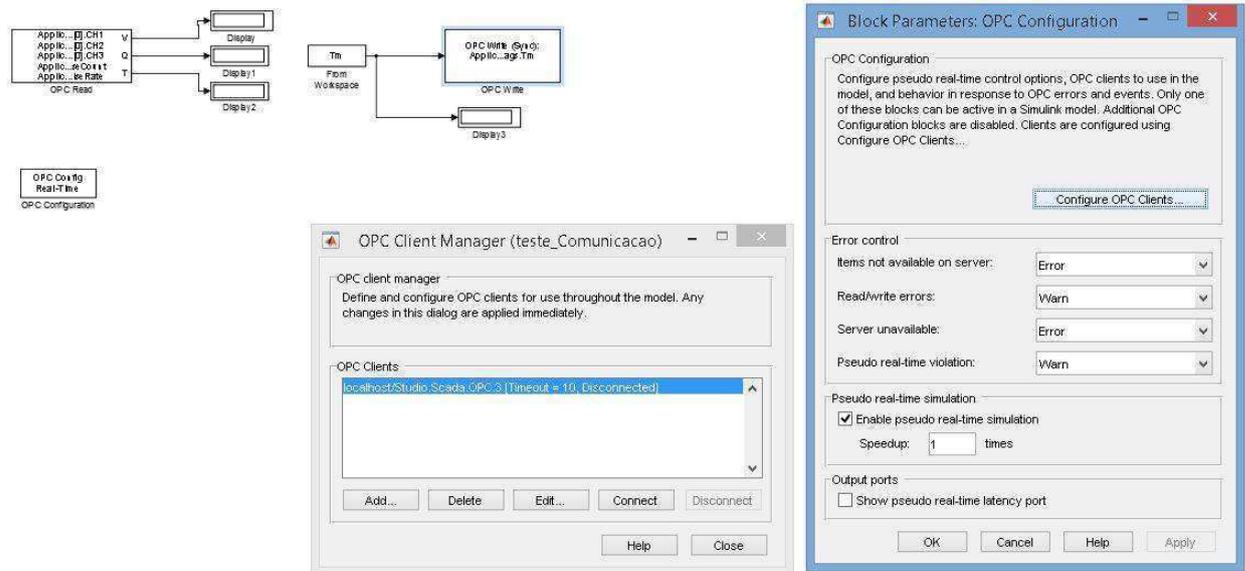
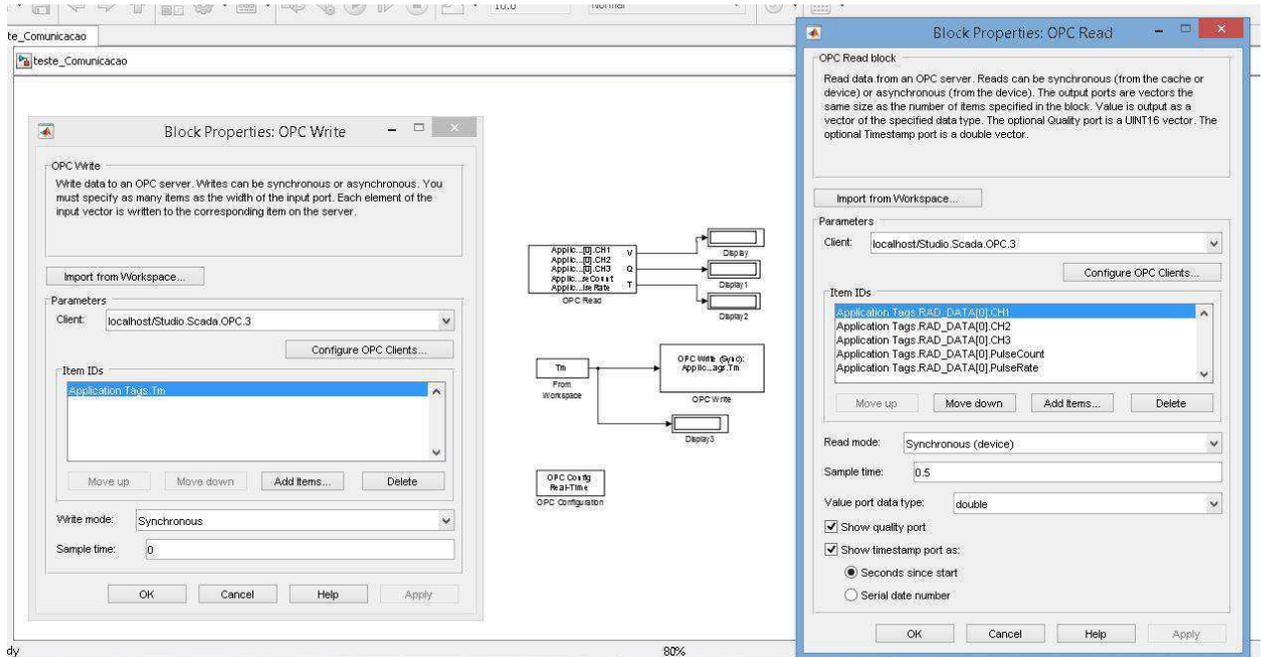


Figura 37: Configuração da comunicação via OPC

Tendo realizado a configuração do protocolo de comunicação OPC, pode-se ler e escrever em Tags do IWS a partir dos blocos *OPC read* e *OPC write*, respectivamente. Estes blocos também podem ser encontrados na biblioteca do Matlab/Simulink OPC toolbox. Para ler uma das Tags existentes no IWS deve-se acrescentar o bloco OPC read ao projeto do Simulink e acessar as propriedades desse bloco. Na janela de propriedades é necessário escolher o servidor no qual desejamos coletar os dados no campo *Client*, haja vista que um OPC cliente pode ter diversos OPC servidor. Tendo feito isto, podemos ler qualquer Tag existente no IWS clicando no botão *Add Items...*, e adicionado as Tags desejadas.

O mesmo processo deve ser realizado se quisermos escrever em uma Tag do IWS. No bloco *OPC write*, deve-se escolher o servidor relativo ao equipamento que se deseja acessar os dados, e em seguida adicionar as variáveis desejadas em *Add Items...* Desta forma podemos escrever qualquer tipo de dados nas Tags do IWS. Estes procedimentos podem ser verificados na figura 36.



**Figura 38:** Leitura e escrita de Tags do IWS

Como o algoritmo de localização foi desenvolvido no Matlab, *arquivo.m*, existe a necessidade de realizar a troca de dados entre o Matlab e o Simulink. Esta permuta de dados pode ser realizada a partir de dois blocos existentes na biblioteca do Simulink, o *FromWorkspace*, responsável por receber dados do workspace do Matlab, e o *ToWorkspace*, responsável por enviar os dados para o Matlab. Desta forma, podemos enviar as Tags necessárias para o algoritmo de localização, e receber os dados necessários para que o IWS possa representar essa localização graficamente. As tags que precisam ser enviadas ao algoritmo de localização são as posições x, y dos radiômetros, de acordo com a origem definida na tela de configurações do sistema, e a localização do transmissor utilizado para calibração dos radiômetros, também definido na tela de configuração.

## 5. Resultados e Considerações

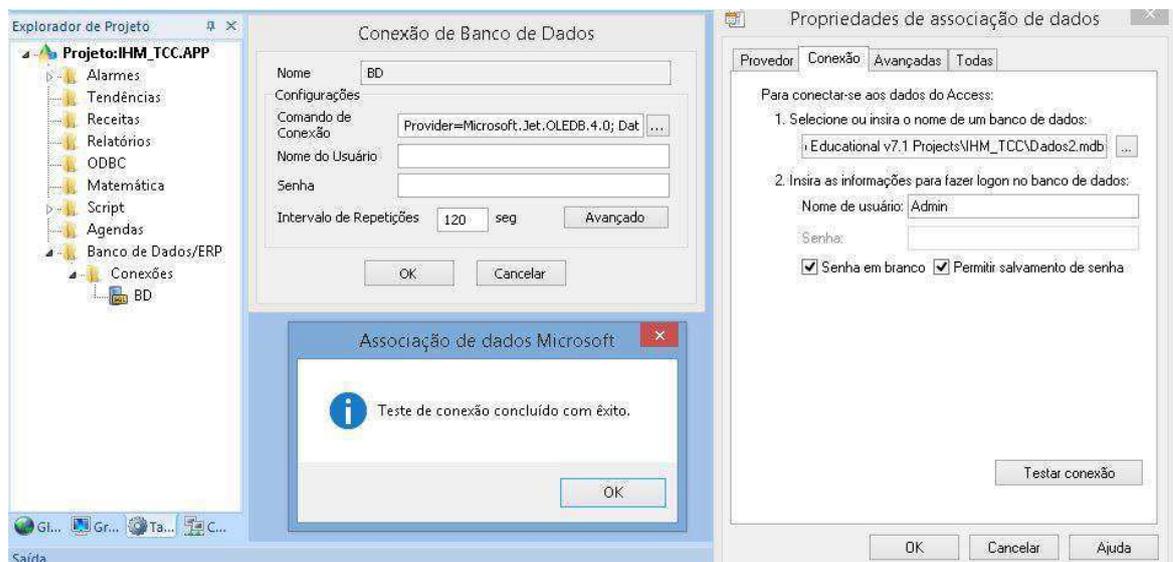
Para validar o funcionamento da IHM projetada foram realizados testes de conexão com o Access e com o Matlab, além da validação do cálculo dos parâmetros para cada radiômetro. Os testes serão descritos a seguir:

### 1. Validação do cálculo dos parâmetros para cada radiômetro

A validação dos cálculos se deu no modo run-time da IHM, ou seja, após a implementação do algoritmo de tratamento dos dados, executamos a IHM e verificamos manualmente se os parâmetros calculados estavam corretos.

### 2. Teste de conexão entre o IWS e o Access

O teste de conexão com o Access é feito de maneira automática e está representado na figura 37.



**Figura 39:** Teste de conexão entre o IWS e o BD

### 3. Teste de conexão entre o IWS e o Matlab/Simulink

A conexão entre o IWS e o Matlab/Simulink é feita através do bloco OPC Configuration, mostrado na seção 4.7 (Comunicação entre Matlab/Simulink e Indusoft). Após configurar este bloco e adicionar os blocos de leitura e escrita de variáveis, é necessário executar a IHM e logo após o programa desenvolvido no Simulink para que a leitura e escrita seja feita.

#### 4. Teste de usabilidade da IHM

O teste de usabilidade da IHM foi realizado pela equipe de desenvolvimento da UFCG e pelo Professor Ian Glover, responsável pela equipe das universidades parceiras. O teste consistiu na utilização da IHM para verificar se os objetivos de usabilidade foram atendidos. Então, verificou-se que os objetivos foram atendidos com sucesso.

### **5.1. Proposta de Testes de Integração**

Sabendo que os testes realizados não são suficientes para certificar que a IHM terá um bom funcionamento quando inserida ao sistema supervisor da subestação, visto que testes com o de conexão com o novo banco de dados, que conterà todos os arquivos de dados coletados por cada radiômetro, devem ser realizados. Alguns testes que foram sugeridos pela equipe da UFCG são descritos abaixo:

1. Teste de conexão entre a IHM e o novo BD
2. Teste de usabilidade realizado pelo operador da subestação
3. Análise de possíveis erros
4. Teste da conexão entre IWS e Matlab/Simulink durante a execução contínua
5. Verificar se a representação dos radiômetros para uma grande quantidade de dados está correta.

Então, só após a realização de todos esses testes e atualização da IHM é que podemos certificar o bom funcionamento da IHM e liberar sua utilização contínua.

### **5.2. Considerações Finais**

Tendo concluído este projeto podemos verificar que todos os objetivos foram atendidos. A interface foi desenvolvida de maneira a possibilitar que o usuário manipulasse todos os recursos sem a necessidade de um conhecimento prévio, desta forma ele pode aprender por tentativa e erro. Um banco de dados para teste foi desenvolvido no Microsoft Access com êxito, já que um BD que comporte todos os arquivos que serão atualizados continuamente pelo

sistema de aquisição de dados será projetado no futuro. A representação de um grande volume de dados a partir da representação visual escolhida atendeu todas as expectativas da equipe de projeto, mas ainda precisa de testes a serem realizados no local de implementação da IHM. E finalmente, a execução do algoritmo de localização das descargas parciais pelo Matlab foi possível graças a realização da comunicação via protocolo OPC.

Com a realização deste projeto podemos concluir que o monitoramento contínuo de equipamentos de uma subestação é de grande valia para as empresas do setor elétrico, visto que em muitas delas este monitoramento é realizado mensalmente, ou até trimestralmente, mas este tipo de monitoramento pode ser tardio caso a incidência de descargas parciais internas a um dos equipamentos cresça exponencialmente. Portanto, o sistema de supervisão proposto por este trabalho visa o monitoramento contínuo dos equipamentos através dos dados coletados por cada radiômetro, desta forma o operador poderá verificar a incidência das descargas parciais em um dos radiômetros, e apoiado pelo sistema de representação visual abordado tomar a decisão correta.

Em relação ao conteúdo assimilado durante o desenvolvimento deste trabalho, inclui conhecimento de um novo software de desenvolvimento de interfaces gráficas, aprendizado da linguagem de manipulação de dados SQL e da linguagem VBScript, aperfeiçoamento do uso do MCIE, conhecimento do método para conexão do IWS com outros dispositivos, entre outros conhecimentos adquiridos ao longo do projeto. Por fim, o desenvolvimento deste projeto proporcionou a interação com equipes de universidades estrangeiras e com uma alta carga de conhecimento a ser compartilhado.

A IHM alcançou os objetivos propostos, mas ainda é necessário:

- Desenvolvimento de um Banco de Dados
- Integração do Software Supervisório com o BD
- Representação da localização das DPs na IHM
- Validar a representação dos radiômetros junto aos usuários finais, testando a usabilidade com um operador da subestação
- Realizar os ajustes necessários.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Neto, J. M.; Zhang, Y.; Jaber, A.; Zhu, M.; Judd, M.; Atkinson, R.; Soraghan, J.; Neto, J. S.; Vieira, M. F. Q.; Glover, I. A.; "Radiometric Location of Partial Discharge Sources for the Future Smart Grid".
- [2] Jaber, A.; Lazaridis, P.; Zhang, Y.; Upton, D.; Ahmed, H.; Khan, U.; Saeed, B.; "Comparison of Contact Measurement and Free-Space Radiation Measurement of Partial Discharge Signals".
- [3] S. Electric, "VBScript Reference Manual for InduSoft Web Studio," [Online]. Available: [www.indusoft.com/pdf/VBScript%20Reference.pdf](http://www.indusoft.com/pdf/VBScript%20Reference.pdf). [Acesso em 05 02 2016].
- [4] E. J. T. Junior, "Medição de descargas parciais," [Online]. Available: [www.eletrica.ufpr.br/piazza/ensaios/meddp1.pdf](http://www.eletrica.ufpr.br/piazza/ensaios/meddp1.pdf). [Acesso em 10 02 2016].
- [5] S. Electric, "Indusoft Web Studio," Schneider Electric, [Online]. Available: <http://www.indusoft.com/br/>. [Acesso em 26 01 2016].
- [6] M. F. Q. Vieira, "Notas de aula da disciplina Informática Industrial e Método MCIE," [Online]. Available: [https://sites.google.com/a/dee.ufcg.edu.br/fatima/cursos-ministrados/informatica-industrial/downloads\\_infind](https://sites.google.com/a/dee.ufcg.edu.br/fatima/cursos-ministrados/informatica-industrial/downloads_infind). [Acesso em 05 02 2016].
- [7] S. Electric, "Apostila de treinamento Básico - Indusoft Web Studio," [Online]. Available: <http://www.indusoft.com/br/Treinamento>. [Acesso em 02 Dezembro 2015].
- [8] G. Inc., "Google Maps," [Online]. Available: <https://www.google.com.br/maps>. [Acesso em 10 Janeiro 2016].
- [9] J. M. Rosário, "Automação Industrial," Baraúnas, São Paulo, 2009.
- [10] M. FONSECA, "Comunicação OPC – Uma abordagem prática. VI Process Automation Seminar.,” 2002. [Online]. Available: <http://www.cpdee.ufmg.br/~seixas/PaginaSDA/Download/DownloadFiles/OPCMarcosFonseca.PDF>. [Acesso em 30 Maio 2016].
- [11] i. The MathWorks, "MathWorks - Matlab," 1996-2016. [Online]. Available: <http://www.mathworks.com/products/matlab/?requestedDomain=www.mathworks.com>. [Acesso em 20 05 2016].
- [12] S. Electric, "VBscript Reference Manual - Indusoft," [Online]. Available: [www.indusoft.com/pdf/VBScript%20Reference.pdf](http://www.indusoft.com/pdf/VBScript%20Reference.pdf). [Acesso em 05 Fevereiro 2016].
- [13] R. S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill; 3rd ed, 1995.

- [14] Eurovolt, "Monitoramento contínuo de descargas parciais," [Online]. Available: <http://www.eurovolt.com.br/product/monitoramento-contínuo-de-descargas-parciais/>. [Acesso em 20 Março 2016].
- [15] E. Cepel, "Notícias," [Online]. Available: <http://www.cepel.br/sala-de-imprensa/noticias/menu/noticias/cepel-realiza-medicoes-de-descargas-parciais-em-subestacoes-da-eletronorte-em-mato-grosso.htm>. [Acesso em 20 Março 2016].
- [16] Adimarco, "Sistemas de Monitoramento," [Online]. Available: <http://www.adimarco.com.br/produtos/teste-diagnostico-equipamentos-subestacao/sistemas-de-monitoramento/>. [Acesso em 20 Março 2016].
- [17] D.-K. K. B. H. K. Dong-Joo Kang, "Visualization Issues of Mass Data for Efficient HMI Design on Control System in Electric Power Industry," [Online]. Available: <http://www.iasdr2009.or.kr/Papers/Orally%20Presented%20Papers/Interaction/Visualization%20Issues%20of%20Mass%20Data%20for%20Efficient%20HMI%20Design%20on%20Control%20System%20in%20Electric%20Power%20Industry.pdf>. [Acesso em 10 Março 2016].
- [18] B. V.-H. Dorothea Pantförder, "Benefit and evaluation of interactive 3D process data visualization in operator training of plant manufacturing industry," [Online]. Available: [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5346921&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D5346921](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5346921&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5346921). [Acesso em 10 Março 2016].

## ANEXO - CÓDIGOS VBScript e SQL

No presente anexo serão disponibilizados os scripts utilizados neste projeto para realizar o processamento dos dados recebidos e para representação dos radiômetros de acordo com estes dados.

### 1. CRIAÇÃO E INSERÇÃO DE DADOS EM UMA TABELA

'Criação das tabelas referentes a cada radiometro no BD

'RAD3

```
sql = "CREATE TABLE RAD3 (ID int, Date_Time char(50), MotelID Int, Pulse Int, Time_ns Int, Ch1_mV
Decimal(18,4), Ch2_mV Decimal(18,4), Ch3_mV Decimal(18,4))"
```

```
$DBExecute("BD", sql)
```

'RAD7

```
sql = "CREATE TABLE RAD7 (ID int, Date_Time char(50), MotelID Int, Pulse Int, Time_ns Int, Ch1_mV
Decimal(18,4), Ch2_mV Decimal(18,4), Ch3_mV Decimal(18,4))"
```

```
$DBExecute("BD", sql)
```

'RAD8

```
sql = "CREATE TABLE RAD8 (ID int, Date_Time char(50), MotelID Int, Pulse Int, Time_ns Int, Ch1_mV
Decimal(18,4), Ch2_mV Decimal(18,4), Ch3_mV Decimal(18,4))"
```

```
$DBExecute("BD", sql)
```

\*\*\*\*\*

'INSERINDO DADOS

```
sql = "INSERT INTO RAD3 (ID, Date_Time, MotelID, Pulse, Time_ns, Ch1_mV, Ch2_mV, Ch3_mV) SELECT ID,
Date_Time, MotelID, Pulse, Time_ns, Ch1_mV, Ch2_mV, Ch3_mV FROM Dados1 WHERE MotelID = 3; "
```

```
$DBExecute("BD", sql)
```

```
sql = "INSERT INTO RAD7 (ID, Date_Time, MotelID, Pulse, Time_ns, Ch1_mV, Ch2_mV, Ch3_mV) SELECT ID,
Date_Time, MotelID, Pulse, Time_ns, Ch1_mV, Ch2_mV, Ch3_mV FROM Dados1 WHERE MotelID = 7; "
```

```
$DBExecute("BD", sql)
```

```
sql = "INSERT INTO RAD8 (ID, Date_Time, MotelID, Pulse, Time_ns, Ch1_mV, Ch2_mV, Ch3_mV) SELECT ID,
Date_Time, MotelID, Pulse, Time_ns, Ch1_mV, Ch2_mV, Ch3_mV FROM Dados1 WHERE MotelID = 8; "
```

```
$DBExecute("BD", sql)
```

## 2. DETERMINAÇÃO DOS PARAMETROS DE CADA RADIOMETRO

### ▪ Determinação dos Parâmetros do Radiômetro 3

'O código configurado aqui é executado enquanto a condição configurada no campo Execução for verdadeira.

'Procedimento para determinação dos parametros de cada radiometro -----

Dim sql, numCursor, numLinhas, Linha, ValorAtualCh1, ValorAtualCh2, ValorAtualCh3,  
ValorAnteriorCh1, ValorAnteriorCh2, ValorAnteriorCh3, N1Reset, N2Reset, N3Reset, TmS

```
sql = "SELECT * FROM RAD3"
```

```
numCursor = $DBCursorOpenSQL("BD", sql)'Ponteiro para a tabela virtual
numLinhas = $DBCursorRowCount(numCursor) 'Contagem do numero de linhas da tabela
```

```
TmS = $Tm / 1000000000 'Time Mesurement em segundos
```

```
For Linha = 1 To numLinhas
```

```
ValorAtualCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
ValorAtualCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
ValorAtualCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")
```

```
If $Tm < $DBCursorGetValue(numCursor, "Time_ns") Then 'Verifica se o
valor do tempo de medição ultrapassou o valor desejado
```

```
$DBCursorClose(numCursor)
```

```
Exit For
```

```
Else
```

```
If Linha > 1 Then
```

```
If ValorAtualCh1 < ValorAnteriorCh1 Then
N1Reset = N1Reset + 1
```

```
End If
```

```
If ValorAtualCh2 < ValorAnteriorCh2 Then
N2Reset = N2Reset + 1
```

```
End If
```

```
If ValorAtualCh3 < ValorAnteriorCh3 Then
N3Reset = N3Reset + 1
```

```
End If
```

```
End If
```

```
End If
```

```
ValorAnteriorCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
ValorAnteriorCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
ValorAnteriorCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")
```

```
$RAD_DATA[0].PulseCount = $DBCursorGetValue(numCursor, "Pulse")
```

'Determinação da variável Pulse Count

```
$RAD_DATA[0].PulseRate = $RAD_DATA[0].PulseCount / TmS
```

'Determinação da variável Pulse Rate

```
$DBCursorNext(numCursor)
```

```
Next
```

'Correção dos valores de potência

\$RAD\_DATA[0].CH1 = N1Reset / TmS

If \$RAD\_DATA[0].CH1 > 0 Then  
     \$RAD\_DATA[0].CH1 = \$RAD\_DATA[0].CH1 + 0.5

End If

\$RAD\_DATA[0].CH2 = N2Reset / TmS

If \$RAD\_DATA[0].CH2 > 0 Then  
     \$RAD\_DATA[0].CH2 = \$RAD\_DATA[0].CH2 + 0.5

End If

\$RAD\_DATA[0].CH3 = N3Reset / TmS

If \$RAD\_DATA[0].CH3 > 0 Then  
     \$RAD\_DATA[0].CH3 = \$RAD\_DATA[0].CH3 + 0.5

End If

\$RAD\_DATA[0].P\_max = \$RAD\_DATA[0].CH1 + \$RAD\_DATA[0].CH2 + \$RAD\_DATA[0].CH3

'Fim do procedimento de determinação dos parametros de cada radiometro

#### ▪ Determinação dos Parâmetros do Radiômetro 7

'O código configurado aqui é executado enquanto a condição configurada no campo Execução for verdadeira.

'Procedimento para determinação dos parametros de cada radiometro -----

Dim sql, numCursor, numLinhas, Linha, ValorAtualCh1, ValorAtualCh2, ValorAtualCh3,  
 ValorAnteriorCh1, ValorAnteriorCh2, ValorAnteriorCh3, N1Reset, N2Reset, N3Reset, TmS

sql = "SELECT \* FROM RAD7"

numCursor = \$DBCursorOpenSQL("BD", sql)'Ponteiro para a tabela virtual  
 numLinhas = \$DBCursorRowCount(numCursor) 'Contagem do numero de linhas da tabela

TmS = \$Tm / 1000000000 'Time Mesurement em segundos

For Linha = 1 To numLinhas

ValorAtualCh1 = \$DBCursorGetValue(numCursor, "Ch1\_mV")  
 ValorAtualCh2 = \$DBCursorGetValue(numCursor, "Ch2\_mV")  
 ValorAtualCh3 = \$DBCursorGetValue(numCursor, "Ch3\_mV")

If \$Tm < \$DBCursorGetValue(numCursor, "Time\_ns") Then 'Verifica se o  
 valor do tempo de medição ultrapassou o valor desejado

    \$DBCursorClose(numCursor)

    Exit For

Else

    If Linha > 1 Then

        If ValorAtualCh1 < ValorAnteriorCh1 Then

            N1Reset = N1Reset + 1

```

End If
If ValorAtualCh2 < ValorAnteriorCh2 Then
    N2Reset = N2Reset + 1
End If
If ValorAtualCh3 < ValorAnteriorCh3 Then
    N3Reset = N3Reset + 1
End If
End If

End If

ValorAnteriorCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
ValorAnteriorCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
ValorAnteriorCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")

$RAD_DATA[1].PulseCount = $DBCursorGetValue(numCursor, "Pulse")
'Determinação da variável Pulse Count
$RAD_DATA[1].PulseRate = $RAD_DATA[1].PulseCount / TmS
'Determinação da variável Pulse Rate

$DBCursorNext(numCursor)
Next

'Correção dos valores de potência

$RAD_DATA[1].CH1 = N1Reset / TmS

If $RAD_DATA[1].CH1 > 0 Then
    $RAD_DATA[1].CH1 = $RAD_DATA[1].CH1 + 0.5
End If

$RAD_DATA[1].CH2 = N2Reset / TmS

If $RAD_DATA[1].CH2 > 0 Then
    $RAD_DATA[1].CH2 = $RAD_DATA[1].CH2 + 0.5
End If

$RAD_DATA[1].CH3 = N3Reset / TmS

If $RAD_DATA[1].CH3 > 0 Then
    $RAD_DATA[1].CH3 = $RAD_DATA[1].CH3 + 0.5
End If

$RAD_DATA[1].P_max = $RAD_DATA[1].CH1 + $RAD_DATA[1].CH2 + $RAD_DATA[1].CH3

```

#### ▪ Determinação dos Parâmetros do Radiômetro 8

'O código configurado aqui é executado enquanto a condição configurada no campo Execução for verdadeira.

'Procedimento para determinação dos parametros de cada radiometro -----

Dim sql, numCursor, numLinhas, Linha, ValorAtualCh1, ValorAtualCh2, ValorAtualCh3,  
ValorAnteriorCh1, ValorAnteriorCh2, ValorAnteriorCh3, N1Reset, N2Reset, N3Reset, TmS

```
sql = "SELECT * FROM RAD8"
```

```

numCursor = $DBCursorOpenSQL("BD", sql)'Ponteiro para a tabela virtual
numLinhas = $DBCursorRowCount(numCursor) 'Contagem do numero de linhas da tabela

TmS = $Tm / 1000000000      'Time Mesurement em segundos

For Linha = 1 To numLinhas

    ValorAtualCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
    ValorAtualCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
    ValorAtualCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")

    If $Tm < $DBCursorGetValue(numCursor, "Time_ns") Then      'Verifica se o
valor do tempo de medição ultrapassou o valor desejado
        $DBCursorClose(numCursor)
        Exit For
    Else
        If Linha > 1 Then
            If ValorAtualCh1 < ValorAnteriorCh1 Then
                N1Reset = N1Reset + 1
            End If
            If ValorAtualCh2 < ValorAnteriorCh2 Then
                N2Reset = N2Reset + 1
            End If
            If ValorAtualCh3 < ValorAnteriorCh3 Then
                N3Reset = N3Reset + 1
            End If
        End If
    End If

    ValorAnteriorCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
    ValorAnteriorCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
    ValorAnteriorCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")

    $RAD_DATA[2].PulseCount = $DBCursorGetValue(numCursor, "Pulse")
'Determinação da variável Pulse Count
    $RAD_DATA[2].PulseRate = $RAD_DATA[2].PulseCount / TmS
        'Determinação da variável Pulse Rate

    $DBCursorNext(numCursor)
Next

'Correção dos valores de potência

$RAD_DATA[2].CH1 = N1Reset / TmS

If $RAD_DATA[2].CH1 > 0 Then
    $RAD_DATA[2].CH1 = $RAD_DATA[2].CH1 + 0.5
End If

$RAD_DATA[2].CH2 = N2Reset / TmS

If $RAD_DATA[2].CH2 > 0 Then
    $RAD_DATA[2].CH2 = $RAD_DATA[2].CH2 + 0.5
End If

$RAD_DATA[2].CH3 = N3Reset / TmS

If $RAD_DATA[2].CH3 > 0 Then

```

```
$RAD_DATA[2].CH3 = $RAD_DATA[2].CH3 + 0.5
```

```
End If
```

```
$RAD_DATA[2].P_max = $RAD_DATA[2].CH1 + $RAD_DATA[2].CH2 + $RAD_DATA[2].CH3
```

### 3. REPRESENTAÇÃO DOS RADIÔMETROS (RAIO E SETORES DAS CIRCUNFERÊNCIAS)

#### ▪ Representação do Radiômetro 3

'O código configurado aqui é executado enquanto a condição configurada no campo Execução for verdadeira.

'Procedimento para determinação dos parametros de cada radiometro -----

```
Dim sql, numCursor, numLinhas, Linha, ValorAtualCh1, ValorAtualCh2, ValorAtualCh3,
ValorAnteriorCh1, ValorAnteriorCh2, ValorAnteriorCh3, N1Reset, N2Reset, N3Reset, TmS_max
```

```
sql = "SELECT * FROM RAD3"
```

```
numCursor = $DBCursorOpenSQL("BD", sql)'Ponteiro para a tabela virtual
numLinhas = $DBCursorRowCount(numCursor) 'Contagem do numero de linhas da tabela
```

```
For Linha = 1 To numLinhas
```

```
ValorAtualCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
ValorAtualCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
ValorAtualCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")
```

```
    If Linha > 1 Then
```

```
        If ValorAtualCh1 < ValorAnteriorCh1 Then
            N1Reset = N1Reset + 1
```

```
        End If
```

```
        If ValorAtualCh2 < ValorAnteriorCh2 Then
            N2Reset = N2Reset + 1
```

```
        End If
```

```
        If ValorAtualCh3 < ValorAnteriorCh3 Then
            N3Reset = N3Reset + 1
```

```
        End If
```

```
    End If
```

```
ValorAnteriorCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
ValorAnteriorCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
ValorAnteriorCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")
```

```
$Rep_Rad[0].PulseCount = $DBCursorGetValue(numCursor, "Pulse")
```

'Determinação da variável Pulse Count

```
$DBCursorNext(numCursor)
Next
```

```
TmS_max = $DBCursorGetValue(numCursor, "Time_ns") / 1000000000
```

```
$Rep_Rad[0].PulseRate = $Rep_Rad[0].PulseCount / TmS_max
'Determinação da variável Pulse Rate
```

```
'Correção dos valores de potência
```

```
$Rep_Rad[0].P1_max = N1Reset / TmS_max
```

```
If $Rep_Rad[0].P1_max > 0 Then
    $Rep_Rad[0].P1_max = $Rep_Rad[0].P1_max + 0.5
End If
```

```
$Rep_Rad[0].P2_max = N2Reset / TmS_max
```

```
If $Rep_Rad[0].P2_max > 0 Then
    $Rep_Rad[0].P2_max = $Rep_Rad[0].P2_max + 0.5
End If
```

```
$Rep_Rad[0].P3_max = N3Reset / TmS_max
```

```
If $Rep_Rad[0].P3_max > 0 Then
    $Rep_Rad[0].P3_max = $Rep_Rad[0].P3_max + 0.5
End If
```

```
'Raio dos radiometros
```

```
$Rep_Rad[0].P_max = $Rep_Rad[0].P1_max + $Rep_Rad[0].P2_max + $Rep_Rad[0].P3_max
```

```
If $Rep_Rad[0].P_max = 0 Then
    $Rep_Rad[0].rSize = 0
```

```
End If
```

```
If $Rep_Rad[0].P_max > 0 Then
    $Rep_Rad[0].rSize = $RAD_DATA[0].P_max*100 / $Rep_Rad[0].P_max
```

```
End If
```

```
'Setores dos radiometros
```

```
If $Rep_Rad[0].P_max = 0 Then
    $V_rad3[0] = 0
    $V_rad3[1] = 0
    $V_rad3[2] = 0
```

```
End If
```

```
If $Rep_Rad[0].P_max > 0 Then
    $V_rad3[0] = (($RAD_DATA[0].CH1) / $Rep_Rad[0].P_max)*100
    $V_rad3[1] = (($RAD_DATA[0].CH2) / $Rep_Rad[0].P_max)*100
    $V_rad3[2] = (($RAD_DATA[0].CH3) / $Rep_Rad[0].P_max)*100
```

```
End If
```

```
'Fim do procedimento de determinação dos parametros de cada radiometro
```

## ▪ Representação do Radiômetro 7

'O código configurado aqui é executado enquanto a condição configurada no campo Execução for verdadeira.

'Procedimento para determinação dos parametros de cada radiometro -----

Dim sql, numCursor, numLinhas, Linha, ValorAtualCh1, ValorAtualCh2, ValorAtualCh3,  
ValorAnteriorCh1, ValorAnteriorCh2, ValorAnteriorCh3, N1Reset, N2Reset, N3Reset, TmS\_max

```
sql = "SELECT * FROM RAD7"
```

```
numCursor = $DBCursorOpenSQL("BD", sql)'Ponteiro para a tabela virtual
```

```
numLinhas = $DBCursorRowCount(numCursor) 'Contagem do numero de linhas da tabela
```

```
For Linha = 1 To numLinhas
```

```
ValorAtualCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
```

```
ValorAtualCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
```

```
ValorAtualCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")
```

```
    If Linha > 1 Then
```

```
        If ValorAtualCh1 < ValorAnteriorCh1 Then
            N1Reset = N1Reset + 1
```

```
        End If
```

```
        If ValorAtualCh2 < ValorAnteriorCh2 Then
            N2Reset = N2Reset + 1
```

```
        End If
```

```
        If ValorAtualCh3 < ValorAnteriorCh3 Then
            N3Reset = N3Reset + 1
```

```
        End If
```

```
    End If
```

```
ValorAnteriorCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
```

```
ValorAnteriorCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
```

```
ValorAnteriorCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")
```

```
$Rep_Rad[1].PulseCount = $DBCursorGetValue(numCursor, "Pulse")
```

'Determinação da variável Pulse Count

```
$DBCursorNext(numCursor)
```

```
Next
```

```
TmS_max = $DBCursorGetValue(numCursor, "Time_ns") / 1000000000
```

```
$Rep_Rad[1].PulseRate = $Rep_Rad[1].PulseCount / TmS_max
```

'Determinação da variável Pulse Rate

'Correção dos valores de potência

```
$Rep_Rad[1].P1_max = N1Reset / TmS_max
```

```
If $Rep_Rad[1].P1_max > 0 Then
```

```
    $Rep_Rad[1].P1_max = $Rep_Rad[1].P1_max + 0.5
```

```
End If
```

```

$Rep_Rad[1].P2_max = N2Reset / TmS_max

If $Rep_Rad[1].P2_max > 0 Then
    $Rep_Rad[1].P2_max = $Rep_Rad[1].P2_max + 0.5
End If

$Rep_Rad[1].P3_max = N3Reset / TmS_max

If $Rep_Rad[1].P3_max > 0 Then
    $Rep_Rad[1].P3_max = $Rep_Rad[1].P3_max + 0.5
End If

'Raio dos radiometros
$Rep_Rad[1].P_max = $Rep_Rad[1].P1_max + $Rep_Rad[1].P2_max + $Rep_Rad[1].P3_max

If $Rep_Rad[1].P_max = 0 Then
    $Rep_Rad[1].rSize = 0
End If
If $Rep_Rad[1].P_max > 0 Then
    $Rep_Rad[1].rSize = $RAD_DATA[1].P_max*100 / $Rep_Rad[1].P_max
End If

'Setores dos radiometros
If $Rep_Rad[1].P_max = 0 Then
    $V_rad7[0] = 0
    $V_rad7[1] = 0
    $V_rad7[2] = 0
End If

If $Rep_Rad[1].P_max > 0 Then
    $V_rad7[0] = (($RAD_DATA[1].CH1) / $Rep_Rad[1].P_max)*100
    $V_rad7[1] = (($RAD_DATA[1].CH2) / $Rep_Rad[1].P_max)*100
    $V_rad7[2] = (($RAD_DATA[1].CH3) / $Rep_Rad[1].P_max)*100
End If

'Fim do procedimento de determinação dos parametros de cada radiometro

```

### ▪ Representação do Radiômetro 8

O código configurado aqui é executado enquanto a condição configurada no campo Execução for verdadeira.

'Procedimento para determinação dos parametros de cada radiometro -----

```

Dim sql, numCursor, numLinhas, Linha, ValorAtualCh1, ValorAtualCh2, ValorAtualCh3,
ValorAnteriorCh1, ValorAnteriorCh2, ValorAnteriorCh3, N1Reset, N2Reset, N3Reset, TmS_max

```

```

    sql = "SELECT * FROM RAD8"

```

```

numCursor = $DBCursorOpenSQL("BD", sql)'Ponteiro para a tabela virtual
numLinhas = $DBCursorRowCount(numCursor) 'Contagem do numero de linhas da tabela

```

```

For Linha = 1 To numLinhas

```

```

ValorAtualCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
ValorAtualCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
ValorAtualCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")

    If Linha > 1 Then
        If ValorAtualCh1 < ValorAnteriorCh1 Then
            N1Reset = N1Reset + 1
        End If
        If ValorAtualCh2 < ValorAnteriorCh2 Then
            N2Reset = N2Reset + 1
        End If
        If ValorAtualCh3 < ValorAnteriorCh3 Then
            N3Reset = N3Reset + 1
        End If
    End If

    ValorAnteriorCh1 = $DBCursorGetValue(numCursor, "Ch1_mV")
    ValorAnteriorCh2 = $DBCursorGetValue(numCursor, "Ch2_mV")
    ValorAnteriorCh3 = $DBCursorGetValue(numCursor, "Ch3_mV")

    $Rep_Rad[2].PulseCount = $DBCursorGetValue(numCursor, "Pulse")
'Determinação da variável Pulse Count

    $DBCursorNext(numCursor)
Next

TmS_max = $DBCursorGetValue(numCursor, "Time_ns") / 1000000000

$Rep_Rad[2].PulseRate = $Rep_Rad[2].PulseCount / TmS_max
'Determinação da variável Pulse Rate

'Correção dos valores de potência

$Rep_Rad[2].P1_max = N1Reset / TmS_max

If $Rep_Rad[2].P1_max > 0 Then
    $Rep_Rad[2].P1_max = $Rep_Rad[2].P1_max + 0.5
End If

$Rep_Rad[2].P2_max = N2Reset / TmS_max

If $Rep_Rad[2].P2_max > 0 Then
    $Rep_Rad[2].P2_max = $Rep_Rad[2].P2_max + 0.5
End If

$Rep_Rad[2].P3_max = N3Reset / TmS_max

If $Rep_Rad[2].P3_max > 0 Then
    $Rep_Rad[2].P3_max = $Rep_Rad[2].P3_max + 0.5
End If

'Raio dos radiômetros
$Rep_Rad[2].P_max = $Rep_Rad[2].P1_max + $Rep_Rad[2].P2_max + $Rep_Rad[2].P3_max

If $Rep_Rad[2].P_max = 0 Then
    $Rep_Rad[2].rSize = 0
End If

```

```
If $Rep_Rad[2].P_max > 0 Then
    $Rep_Rad[2].rSize = $RAD_DATA[2].P_max*100 / $Rep_Rad[2].P_max
End If
```

'Setores dos radiômetros

```
If $Rep_Rad[2].P_max = 0 Then
    $V_rad8[0] = 0
    $V_rad8[1] = 0
    $V_rad8[2] = 0
End If
```

```
If $Rep_Rad[2].P_max > 0 Then
    $V_rad8[0] = (($RAD_DATA[2].CH1) / $Rep_Rad[2].P_max)*100
    $V_rad8[1] = (($RAD_DATA[2].CH2) / $Rep_Rad[2].P_max)*100
    $V_rad8[2] = (($RAD_DATA[2].CH3) / $Rep_Rad[2].P_max)*100
End If
```

'Fim do procedimento de determinação dos parâmetros de cada radiômetro

#### 4. DEFINIÇÃO DO DELTA (Potência do Novo Arquivo de Dados em Relação ao Antigo)

- **Radiômetro 3 (Radiômetros 7 e 8 são análogos)**

'O código configurado aqui é executado enquanto a condição configurada no campo Execução for verdadeira.

' $\$delta\_rad3 = (\$rSize\_novo - \$Rep\_Rad[0].rSize) / \$Rep\_Rad[0].rSize$

```
If($delta_rad3 <= 0.05) Then
    $V_rad3[3] = 10
    $V_rad3[4] = 0
    $V_rad3[5] = 0
    $V_rad3[6] = 0
    $V_rad37 = 0
    $V_rad38 = 0
    $V_rad39 = 0
```

End If

```
If($delta_rad3 > 0.05 And $delta_rad3 <= 0.1) Then
    $V_rad3[3] = 0
    $V_rad3[4] = 10
    $V_rad3[5] = 0
    $V_rad3[6] = 0
    $V_rad37 = 0
    $V_rad38 = 0
    $V_rad39 = 0
```

End If

```
If($delta_rad3 > 0.1 And $delta_rad3 <= 0.3) Then
    $V_rad3[3] = 0
    $V_rad3[4] = 0
```

```
$V_rad3[5] = 10  
$V_rad3[6] = 0  
$V_rad37 = 0  
$V_rad38 = 0  
$V_rad39 = 0
```

End If

```
If($delta_rad3 > 0.3 And $delta_rad3 <= 0.5) Then
```

```
$V_rad3[3] = 0  
$V_rad3[4] = 0  
$V_rad3[5] = 0  
$V_rad3[6] = 10  
$V_rad37 = 0  
$V_rad38 = 0  
$V_rad39 = 0
```

End If

```
If($delta_rad3 > 0.5 And $delta_rad3 <= 0.7) Then
```

```
$V_rad3[3] = 0  
$V_rad3[4] = 0  
$V_rad3[5] = 0  
$V_rad3[6] = 0  
$V_rad37 = 10  
$V_rad38 = 0  
$V_rad39 = 0
```

End If

```
If($delta_rad3 > 0.7 And $delta_rad3 <= 1) Then
```

```
$V_rad3[3] = 0  
$V_rad3[4] = 0  
$V_rad3[5] = 0  
$V_rad3[6] = 0  
$V_rad37 = 0  
$V_rad38 = 10  
$V_rad39 = 0
```

End If

```
If($delta_rad3 > 1) Then
```

```
$V_rad3[3] = 0  
$V_rad3[4] = 0  
$V_rad3[5] = 0  
$V_rad3[6] = 0  
$V_rad37 = 0  
$V_rad38 = 0  
$V_rad39 = 10
```

End If